UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

TRABAJO DE FIN DE GRADO

Diseño de un sistema de reducción de complejidad para codificación de vídeo

GRADO EN INGENIERIA DE SISTEMAS AUDIOVISUALES

Autor

Roberto Aldeán Macharé

Tutora

Amaya Jiménez Moreno

Madrid, 2016

RESUMEN

Actualmente, la gran popularidad alcanzada por las aplicaciones que emplean servicios de vídeo, como las videoconferencias, o la creciente demanda de la alta definición, ya sea para la televisión o para dispositivos móviles, ha producido que la codificación de vídeo se convierta en una herramienta vital en el desarrollo de las nuevas tecnologías con el objetivo de transmitir y almacenar de forma eficiente secuencias de vídeo digital.

Desde la década de los años 90, los organismos de estandarización se han encargado de desarrollar diferentes estándares de codificación de vídeo, el último de ellos se denomina HEVC. Este estándar alcanza una capacidad de compresión superior a todos los estándares previos gracias al uso de un conjunto de técnicas novedosas. No obstante, la utilización de dichas técnicas tiene como inconveniente un incremento relevante de la carga computacional producida en el proceso de codificación.

En relación con este hándicap, en los últimos años se han elaborado numerosos métodos con el fin de reducir la complejidad en el proceso de codificación sin que esto implique un decremento significativo de la calidad visual o la eficiencia de compresión.

En este trabajo se propone un algoritmo cuyo objetivo es reducir la complejidad asociada al estándar HEVC. El método propuesto se basa en dividir el cuadro en diferentes regiones independientes y, tras analizar las propiedades estadísticas específicas de cada región, aplicar un test de hipótesis que nos permitirá decidir el número de recursos de codificación a utilizar en cada una de ellas. Una característica esencial de nuestra propuesta será su capacidad de adaptación a lo largo del proceso de codificación a las características intrínsecamente variables de las secuencias de vídeos, teniendo en cuenta la variedad de contenido que podremos encontrar en secuencias diferentes e, incluso, los cambios habituales de contenido a lo largo de una misma secuencia. Con todo ello, puede reducirse la complejidad del codificador de vídeo de forma notable, sin producir grandes pérdidas de eficiencia de compresión.

El método propuesto ha sido examinado de manera exhaustiva y se ha comparado con otros sistemas del estado del arte, empleando para ello un conjunto formado por una gran cantidad de secuencias de contenido diverso y diferentes resoluciones, comprobando que obtiene mejores resultados que los métodos comparados.

ABSTRACT

Currently, the great popularity of applications employing video services such as video conferencing, or the increasing demand for high definition, either for television or for mobile devices, has produced the video encoding becomes a tool essential in the development of new technologies to transmit and efficiently store digital video sequences.

Since the early 90s, standardization organizations have been commissioned to develop different video coding standards, the last one is called HEVC. This standard achieves higher compression capacity above all previous standards through the use of a set of novel techniques. However, the use of these techniques has the drawback a relevant increase in the computational load produced in the encoding process.

In connection with this issue, in recent years numerous methods have been developed in order to reduce complexity in the encoding process without implying a significant decrease in visual quality and compression efficiency.

This paper presents an algorithm which aims to reduce the complexity associated with HEVC standard. The proposed method is based on dividing the frame into several independent regions and, after analyzing the specific statistical properties of each region, apply a hypothesis test that will allow us to decide the number of coding resources to use in each region. An essential feature of our proposal will be its ability to adapt along the coding process to the inherently variable characteristics of video sequences, taking into account the variety of content that can be found in different sequences and even the usual changes included along one sequence. With all of this, the complexity of the video encoder can be reduce significantly, without causing great loss of compression efficiency.

The proposed method has been examined thoroughly and compared with other systems of the state of the art, employing a set consisting of a large number of sequences of different contents and different resolutions, ensuring that outperforms the methods compared.

ÍNDICE

ÍNDICE DE FIGURAS 6
ÍNDICE DE TABLAS 8
ACRÓNIMOS 9
1. INTRODUCTION AND MOTIVATION 13
2. CODIFICACIÓN DE VÍDEO 17
2.1 Introducción 17
2.2 Resolución espacial y temporal 17
2.3 Codificación predictiva diferencial 19
2.3.1 Predicción espacial 20
2.3.2 Predicción temporal 21
2.4 El modelo híbrido DPCM/DCT 22
2.4.1 Proceso de codificación 23
2.4.2 Proceso de decodificación 28
2.5 Tipos de plano 29
3. ESTÁNDAR DE CODIFICACIÓN HEVC 32
3.1 Introducción 32
3.2 Diagrama de bloques HEVC 33
3.3 Estructura de árbol 33
3.4 Predicción Intra 35
3.5 Predicción Inter 36
3.5.1 Modos de visión 36
3.5.2 Interpolación de muestras fraccionarias 37
3.5.3 Merge mode 38
3.5.4 Predicción del vector de movimiento 39
3.6 Transformación 40
3.7 Cuantificación 41
3.8 Codificación entrópica 43
3.9 Optimización tasa-distorsión 45
4. ESTADO DEL ARTE 50
4.1 Introducción 50

4.2 Revisión del estado del arte 50
5. MÉTODO PROPUESTO 56
5.1 Introducción 56
5.2 Test de hipótesis 57
5.3 Variables de entrada 60
5.4 Estimación de los parámetros estadísticos 63
5.5 Regionalización 65
6. RESULTADOS 70
6.1 Configuración 70
6.2 Reducción de complejidad 70
6.3 Comparación con otros métodos del estado del arte 72
7. CONCLUSIONS AND FUTURE LINES OF RESEARCH 79
7.1 Conclusions 79
7.2 Future lines of research 80
8. PLANIFICACIÓN Y PRESPUESTO 83
8.1 Planificación 83
8.2 Presupuesto 83
9. REFERENCIAS 87

ÍNDICE DE FIGURAS

Figura 2.1 Muestreo en el dominio temporal y espacial 17

Figura 2.2 Orden de codificación de un plano 19

Figura 2.3 Estimación de movimiento 20

Figura 2.4 Vectores de movimiento 21

Figura 2.5 Codificador de vídeo híbrido DPCM/DCT 22

Figura 2.6 Decodificador de vídeo híbrido DPCM/DCT 22

Figura 2.7 División en bloques de píxeles 23

Figura 2.8 Ejemplo DCT aplicada sobre un bloque 24

Figura 2.9 Cuantificador uniforme 25

Figura 2.10 Patrón IP 28

Figura 3.1 Diagrama de bloques de un codificador de vídeo HEVC 32

Figura 3.2 División de un CTB aplicando la estructura en forma de árbol 33

Figura 3.3 Direcciones de predicción para las PUs de tipo Intra 34

Figura 3.4 Modos de división de una CU en la predicción Inter 35

Figura 3.5 Interpolación de muestras cuya resolución es fraccional 37

Figura 3.6 Posiciones de candidatos espaciales en el *merge mode* 38

Figura 3.7 Ejemplo de división en Pus y TUs 40

Figura 3.8 Diagrama de CABAC 42

Figura 3.9 Escaneo diagonal (a), horizontal (b) y vertical (c) 43

Figura 5.1 Diagrama de flujo 58

Figura 5.2 Pdf del coste J_{min} asociado a la profundidad 0 para la secuencia *Blowing Bubbles* con QP 32 60

Figura 5.3 Pdf del coste J_{min} asociado a la profundidad 0 para la secuencia *Johnny* con QP 27 60

Figura 5.4 Estimación de $\hat{\mu}_0$ empleando los métodos aritmético y exponencial 63

Figura 5.5 Zonas claramente diferenciadas en un plano 65

Figura 5.6 División en 3 regiones horizontales (a), en cruz (b) y en 4 regiones horizontales (c) 65

Figura 6.1 Rendimiento RD para la secuencia *Johnny* 74

Figura 6.2 Zona ampliada de la figura anterior 74

Figura 6.3 Rendimiento RD para la secuencia *Park Scene* 75

Figura 6.4 Zona ampliada de la figura anterior 75

ÍNDICE DE TABLAS

Tabla 2.1 Resoluciones espaciales comunes 18

Tabla 3.1 Relación entre QP y Q_{step} 41

Tabla 5.1 Medias y desviaciones estándar de los costes RD al seleccionar la hipótesis H_i para la secuencia *Blowing Bubbles* 61

Tabla 5.2 Conjunto reducido de resultados correspondientes a emplear diferentes tipos de regionalización 66

Tabla 6.1 Análisis del rendimiento del método propuesto considerando diferentes tipos de regionalización 70

Tabla 6.2 Análisis del rendimiento del método propuesto en comparación con [Correa et al., 2013] y [Zhang et al., 2013] 72

Tabla 8.1 Coste del personal 82

Tabla 8.2 Coste de los componentes hardware y software 83

ACRÓNIMOS

AVC	Codificación de video avanzada
CABAC	Codificación aritmética binaria adaptada al contexto
CIF	Formato intermedio común
CTB	Bloque de codificación de árbol
CTU	Unidad de codificación de árbol
CU	Unidad de codificación
DCT	Transformada discreta del coseno
DPCM	Modulación por codificación de impulsos diferencial
DVB-S	Broadcast de vídeo digital por satélite
HEVC	Codificación de video de alta eficiencia
IDCT	Transformada inversa discreta del coseno
ISO-MPEG	Organización nacional de estandarización – Grupo de expertos de imágenes en movimiento
ITU	Unión internacional de comunicaciones
ITU- VECG	Unión internacional de comunicaciones – Grupo de expertos de codificación de vídeo
LRT	Test de razón de verosimilitudes
MD	Decisión de modo
PDF	Función densidad de probabilidad
PU	Unidad de predicción
PSNR	Relación señal a ruido de pico
QCIF	Cuarto de CIF
QP	Parámetro de cuantificación

RDO	Optimización tasa-distorsión
TDT	Televisión digital terrestre
TU	Unidad de transformación
UHDTV	Televisión de ultra alta definición
URQ	Cuantificación de reconstrucción uniforme
WPP	Procesamiento en paralelo

CHAPTER 1

INTRODUCTION AND MOTIVATION

1. INTRODUCTION AND MOTIVATION

Video coding has become essential to visualize, store and stream digitized video sequences efficiently. This is due to the high demand of multimedia apps that use video services, the big available amount of videos on the net, 3D movies, ultra high definition television (UHDTV) and requests of users for higher resolutions, among other examples. The main purpose of video coding is to reduce the huge amount of data in a video sequence without losing the least amount of visual quality.

There are different reasons that promote the development of video compression of algorithms. In the first place, the networks and processors, despite of their improved capacities over the last years, their lack of ability of working efficiently without compressed video data, especially with the high resolutions and qualities requested by users nowadays. Secondly, mobile devices present restrictions in terms of storage and processing capacities, been the most used currently. Thirdly, thanks to compression algorithms it is possible to use resources efficiently and thus, reduce costs of other related processes.

In the past decades, different video encoding standards have been developed towards conforming new technologies and promoting interoperability among existing devices. Video encoding standards have reached a higher way of understanding by exploiting the existing redundancy of a digital video sequence in two levels: temporal and space. Moreover, they have to meet the International Communication Union (ITU) standards depending on the area: digital video broadcasting by satellite (DBV-S), digital terrestrial TV or mobile communications, among others. For example, those related with transmission power, spectre frequencies, etc.

High Efficiency Video Coding (HEVC) was developed by ITU – T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) towards working with high definition videos. Its main goal is to improve its compression capacity without losing the least amount of visual quality. A good example is the compression rate, which doubles its predecessor standard of H.264/AVC.

HEVC's new and diverse encoding tools are the following: newer codification units, which include different sizes; a codification structure similar to a tree that has never been developed for any video standard before; improvements on the different types of existing predictions and finally, a big amount of tools to parallelize the execution of codification processes, among others.

Although these features allow reaching higher compression efficiency, they also involve greater complexity in the encoder, which means an important computing load and an increase of time in order to encode a video sequence. Thereby, the main purpose of this project is to reduce complexity of video encoding processes without a serious decrease on compression efficiency and quality of encoded sequences.

This project describes an algorithm, which main purpose is to reduce time of coding, in exchange of generating small relevant losses of efficiency in the encoding process. The principal features of this method are the following: to take into account the existence of different and independent regions within every frame that contributes to the development of the system; the usage of an hypothesis test about Gaussian distributions to decide the amount of resources spent in each region; to decide the amount of resources spent in each region the statistical analysis of the characteristics of these regions; the adaptability to the variables of the video sequences; and the compatibility with videos that held different resolutions.

This document is divided in the following order: chapter 2 explains the general process of a video encoding sequence and chapter 3 describes the main tools of the standard process, HEVC. In addition, chapter 4 develops this issue analysing different investigations available in the state of the art. A concrete explanation of the proposed method and a deeper study of the obtained results are found in chapters 5 and 6 respectively. Moreover, chapter 7 shows the conclusions and the future lines of research. Finally, chapter 8 illustrates the planning made in order to develop this project as well as the budget needed for its implementation.

CAPÍTULO 2

CODIFICACIÓN DE VÍDEO

2. CODIFICACIÓN DE VÍDEO

2.1 Introducción

El proceso de codificación de vídeo tiene como objetivo, comprimir el conjunto total de datos que formen una secuencia de vídeo digital, es decir, representar la mayor cantidad de información en el menor espacio posible.

A la hora de comprimir, es necesario considerar las características tanto de la fuente como del destino de la señal. Respecto a la fuente, existe una gran similitud entre un píxel y sus vecinos, ya sea en el mismo plano o en planos diferentes. En lo que concierne al destino, el sistema visual humano es más sensible a las bajas frecuencias, por lo tanto, todo aquello que el sistema visual humano perciba con mayor dificultad, puede codificarse con menos datos.

El capítulo comienza con una descripción de los procesos necesarios para digitalizar una secuencia de vídeo, seguido por una breve explicación del tipo de codificación empleado. Posteriormente, se explicará en detalle el proceso de codificación y decodificación de una secuencia de vídeo digital.

2.2 Resolución espacial y temporal

Una escena visual es continua en el dominio del tiempo y del espacio, por ello, la representación de una escena visual de forma digital incluye un muestreo tanto a nivel temporal, en el que se define un intervalo de tiempo para ir almacenando planos, como a nivel espacial, en el que se establecen una serie de puntos situados de manera vertical y horizontal similar a una cuadrícula. Dicho muestreo puede apreciarse en la figura 2.1.

El vídeo digital es la representación de una escena de vídeo muestreada de forma digital. Cada muestra espacio-temporal está representada por un número o un conjunto de números que describen la luminancia y el color de dicha muestra.

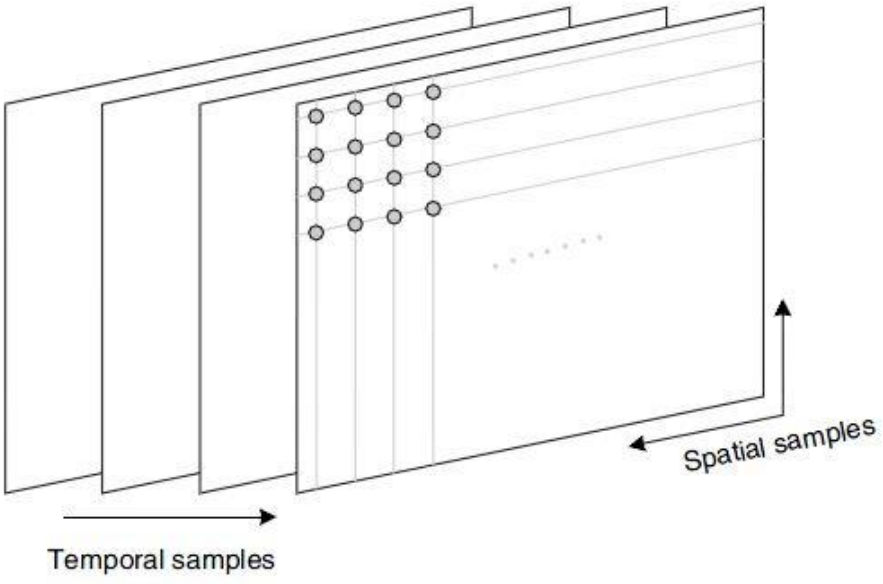

Figura 2.1 Muestreo en el dominio temporal y espacial

En relación con el muestreo espacial, cada muestra espacial se denomina *picture element* (elemento de imagen), lo que se conoce generalmente como píxel. El número de píxeles indica la resolución de la secuencia de vídeo, en la tabla 2.1 se muestran varios formatos de resolución habituales y el número de píxeles de cada una de las líneas verticales y horizontales que componen cada plano. La calidad visual está relacionada de manera directamente proporcional con la resolución espacial, ésta puede variar desde valores muy bajos donde la calidad de la imagen será deficiente, como es el caso del formato de vídeo QCIF, hasta valores muy altos como, por ejemplo, los formatos 4k u 8k que proporcionan una gran definición.

En cuanto al muestreo temporal, el movimiento de una secuencia de vídeo es capturado mediante "instantáneas" realizadas cada cierto intervalo de tiempo definido, denominados planos. Mediante la reproducción de un conjunto de estos planos, se puede lograr la representación del movimiento correspondiente a dicha secuencia de vídeo. Cuanto mayor sea la tasa de muestreo temporal, mejor será la percepción de movimiento, sin embargo, esta mejora de calidad implica un aumento de la cantidad de datos a procesar debido a que se requieren un mayor número de muestras. Las resoluciones temporales más típicas son 25 ó 30 planos por segundo, aunque en el caso de la alta definición pueden llegar a ser 60 ó 120 planos por segundo.

Formato de vídeo	Tamaño del cuadro en píxeles (ancho x alto)
QCIF (*Quarter-CIF*)	176 x 144
CIF (*Common Intermediate Format*)	352 x 288
SD (Standard Definition)	704 x 576
HD (*High Definition*) (720)	1280 x 720
1080	1920 x 1080
2k	2048 x 1536
UHD (4k)	3840 x 2160
4000p	4096 x 3072
Super Hi-Vision (8k)	7680 x 4320

Tabla 2.1 Resoluciones espaciales comunes

2.3 Codificación predictiva diferencial

Debido a la gran cantidad de información que almacenan los planos que componen cada secuencia de vídeo, enviar todos los bits correspondientes a cada uno de estos planos generaría unos valores excesivos de bitrate. Por esta razón se emplea la codificación predictiva diferencial, la cual está basada en construir una predicción de cada plano a partir de las redundancias espaciales y temporales. La diferencia entre la información original y la predicción es habitualmente denominada residuo y será la información que se incluya en la cadena de bits. Siempre que esta predicción se calcule adecuadamente a partir de las redundancias, la energía del residuo será significativamente inferior a la de la información original. Este método se denomina DPCM (*Differential Pulse Code Modulation*) y, gracias a este aspecto fundamental, se consigue una compresión más eficiente.

Existen dos modelos que permiten construir predicciones, el primero a partir de la redundancia espacial y el segundo a partir de la temporal. Si la predicción se basa en información situada alrededor del bloque a codificar y estando presente en el mismo plano, significa que se está haciendo uso de la redundancia espacial y es conocida como predicción Intra. En cambio, si la predicción se basa en información contenida en otros planos previamente codificados, quiere decir que se está explotando la redundancia temporal, lo que se denomina predicción Inter.

Ambos tipos de predicciones serán explicadas con mayor detalle en los siguientes apartados.

2.3.1 Predicción espacial

En cada plano que compone una secuencia de vídeo digital, es muy común encontrar áreas homogéneas o zonas con variaciones muy pequeñas. Esta redundancia espacial, puede emplearse para lograr comprimir la información de manera eficiente. La predicción basada en este concepto es la predicción Intra.

Para poder aplicar este tipo de predicción, es necesario hacer uso de muestras codificadas previamente que correspondan a los píxeles vecinos del bloque que se desea codificar. Debido a que el proceso de codificación sigue un orden de izquierda a derecha y de arriba a abajo, tal y como puede verse en la figura 2.2, se consideran píxeles vecinos aquellos que están situados a la izquierda, arriba y arriba a la izquierda respecto al bloque en cuestión. Se pueden emplear diversas formas a la hora combinar la información contenida en los píxeles vecinos para formar la predicción.

Un concepto muy importante del que se hablará más tarde, es que las muestras empleadas para construir la predicción deben ser muestras reconstruidas, con el fin de evitar la propagación de errores.

El primer plano de todas las secuencias digitales de vídeo es codificado a partir de este tipo de predicción, ya que no existen muestras codificadas correspondientes a planos anteriores.

Figura 2.2 Orden de codificación de un plano

2.3.2 Predicción temporal

En el contenido de vídeo digital, no sólo puede encontrarse redundancia espacial, también puede apreciarse la redundancia temporal ya que la tasa de muestreo temporal es de, al menos, 25 cuadros por segundo. Por ello, una manera inteligente de reducir la cantidad total de bits y lograr una mejor compresión, es hacer uso de muestras ya codificadas pertenecientes a otros planos. A este proceso se le conoce como predicción Inter.

Este tipo de predicción consta de dos pasos, la estimación de movimiento y la compensación de movimiento.

La etapa de estimación de movimiento consiste en buscar en planos codificados anteriormente, el bloque más parecido al que se desea codificar. De esta forma, en lugar de codificar la información contenida en cada uno de los bloques que componen un plano, se codifican los residuos asociados a cada bloque ahorrando así gran cantidad de tasa de bits. Para que la búsqueda del bloque más similar posible pueda tener mayor éxito, se permite realizar dicha búsqueda en varios planos, llamados planos de referencia. Aunque, por otra parte, se restringe el área de búsqueda en dichos planos, ya que el coste computacional que implica es muy elevado. En la figura 2.3 puede apreciarse un ejemplo de esta etapa.

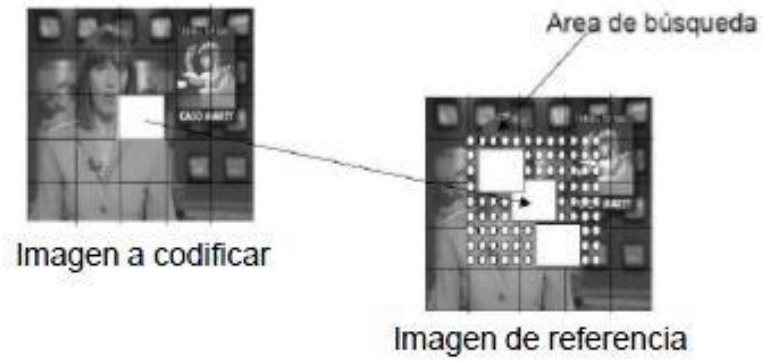

Figura 2.3 Estimación de movimiento

Cuando se haya localizado la posición del bloque más parecido, dicha posición será almacenada en un vector de movimiento, el cual relaciona la posición del bloque

resultado de la búsqueda con la posición del bloque a codificar. En la imagen 2.4 se muestra un ejemplo de dichos vectores.

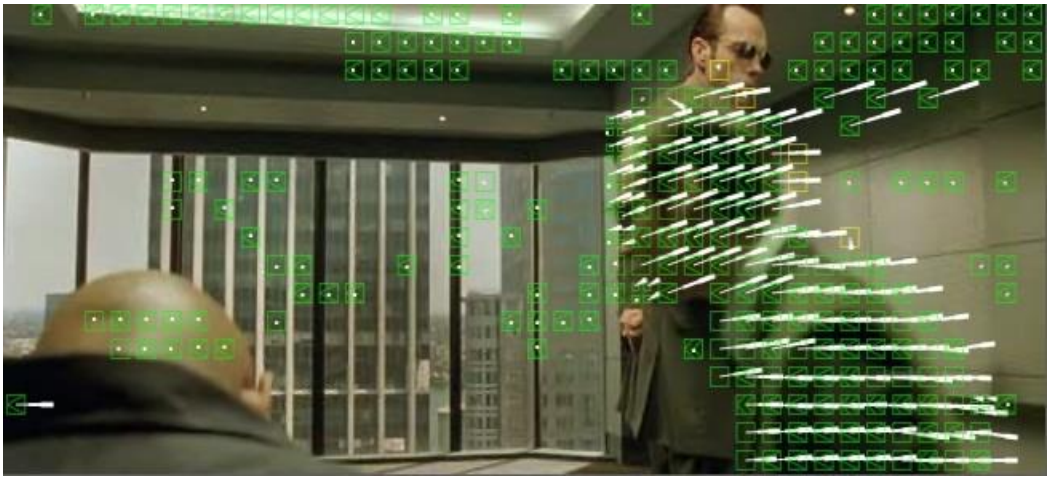

Figura 2.4 Vectores de movimiento

El segundo paso es la compensación de movimiento, cuyo objetivo es identificar el bloque encontrado en el paso anterior y construir con él la predicción. A partir de esta predicción y del bloque original, puede obtenerse el residuo que será codificado finalmente.

Además, es necesario codificar adicionalmente el vector de movimiento y un índice que identifique el plano de referencia, permitiendo así que el decodificador pueda obtener la misma predicción.

2.4 El modelo híbrido DPCM/DCT

La gran mayoría de los estándares de codificación de vídeo, están basados en un prototipo formado por dos bloques principales. En el primer bloque, se lleva a cabo el cálculo del residuo a través de la construcción de una predicción basada en redundancia espacial o temporal, ya indicado anteriormente como DPCM. El segundo bloque se compone de una transformación y un codificador entrópico. A este modelo completo se le denomina generalmente codificador híbrido DPCM/DCT debido a que la transformada más empleada es la transformada discreta del coseno (DCT).

En las figuras 2.5 y 2.6 se muestra un esquema genérico de un codificador y un decodificador híbrido DPCM/DCT respectivamente. En estas figuras, se pueden apreciar los diferentes procesos que atraviesa el plano Fn de una secuencia de vídeo en este modelo híbrido. El codificador se encargará de generar una trama de bits que represente la información del plano Fn de manera comprimida. Por otra parte, el decodificador creará la reconstrucción del plano original, indicado como Fn', a partir de la trama de bits generada por el codificador. Cabe destacar que, por regla general, los planos Fn y Fn' son diferentes debido a pérdidas de información producidas en el proceso de codificación.

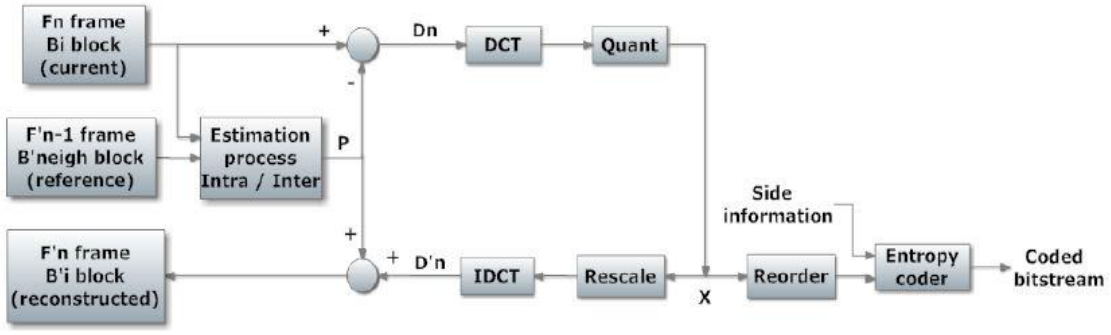

Figura 2.5 Codificador de vídeo híbrido DPCM/DCT

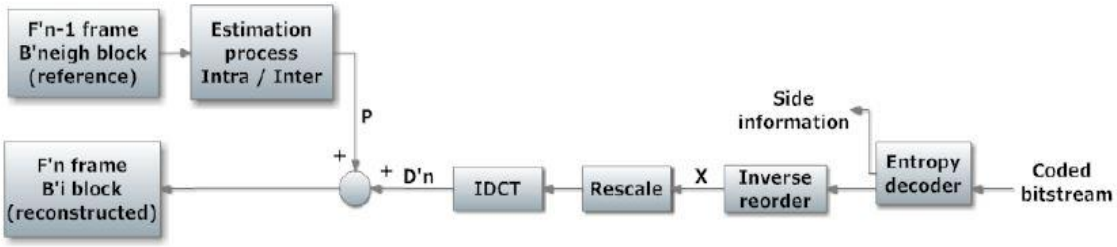

Figura 2.6 Decodificador de vídeo híbrido DPCM/DCT

2.4.1 Proceso de codificación

En primer lugar, se desarrollarán los diferentes pasos a seguir por parte del codificador para obtener la secuencia de bits comprimida que representen la información del plano a codificar. En la figura 2.5, se pueden apreciar dos flujos diferentes, el flujo de

codificación que va de izquierda a derecha, y el flujo de reconstrucción cuyo sentido es de derecha a izquierda.

Las etapas más relevantes del flujo de codificación son las siguientes:

1. El plano a codificar Fn, se divide en pequeñas unidades de trabajo, las cuales están formadas por un bloque de píxeles.

Figura 2.7 División en bloques de píxeles

2. Cada uno de estos bloques es sometido a un proceso para obtener la predicción (P en la figura 2.5), que podrá ser calculada como predicción Intra o Inter. En el caso de predicción Intra se considerarán los píxeles de bloques vecinos para crear la predicción. En el caso de predicción Inter se llevarán a cabo los procesos de estimación y compensación de movimiento.

3. Se procede a calcular el residuo, identificado en la figura 2.5 como Dn, el cual será la diferencia entre el bloque a codificar y la predicción P.

4. El residuo Dn es transformado empleando la DCT. Se recurre a una transformada debido a que en el dominio temporal, la imagen residual es una señal aleatoria, mientras que, en el dominio frecuencial las imágenes están muy polarizadas, es decir, predominan muchas más componentes de baja frecuencia que de altas frecuencias.

Mediante la DCT, la matriz de datos que componen el residuo Dn es transformada en otra matriz, del mismo tamaño, que contiene los valores de las amplitudes, también conocidos como coeficientes, de cada una de las frecuencias que están presentes en dicho residuo. La DCT ordena las componentes espectrales de mayor a menor amplitud, de tal manera que los coeficientes relacionados con las bajas frecuencias al ser más predominantes se concentren al inicio de la matriz, en la esquina superior izquierda, y los coeficientes relacionados con las altas frecuencias se encuentren al final de la matriz, adquiriendo generalmente un valor muy cercano a nulo ya que tienen mucha menor presencia, lo cual facilita la compresión en gran medida.

En la figura 2.8 se muestra el ejemplo de la transformada DCT aplicada a un bloque de 64 x 64 píxeles. A la izquierda de la imagen, está situado el bloque original, mientras que, a la derecha se encuentra el bloque de coeficientes DCT.

Figura 2.8 Ejemplo DCT aplicada sobre un bloque

5. Al bloque de coeficientes transformados, se le aplica un proceso de cuantificación, dando lugar a X en la figura 2.5. Este proceso es un muestreo de los valores de las amplitudes de los coeficientes obtenidos al aplicar la DCT, los cuales únicamente podrán tomar valores entre un conjunto finito de ellos.

   Se ha de aclarar que aplicar la DCT no reduce la cantidad de datos a procesar, por medio de la cuantificación los coeficientes de valores más pequeños tienden a convertirse en nulos, lo que facilita el proceso del codificador entrópico y es clave en el proceso de compresión. Además, en esta etapa se originan las pérdidas de información debido a que, tras cuantificar el valor de los coeficientes, su valor original no puede ser recuperado.

   En la figura 2.9 se muestra un ejemplo de un cuantificador uniforme, en ella se puede apreciar que cuanto mayor sea el número de niveles disponibles de salida del cuantificador, se producirán menores pérdidas ya que el valor

cuantificado será más próximo al valor original y viceversa, cuanto menor sea el número de valores que la salida puede tomar, las pérdidas generadas serán mayores. Sin embargo, cuantos más valores de salida estén disponibles, será necesario un mayor número de bits para representar dichos valores, lo que implica un aumento del flujo de bits a codificar. Por ello, una de las características más relevantes en este apartado, es el escalón de cuantificación, a través del cual se puede controlar la relación entre las pérdidas producidas y el bitrate generado.

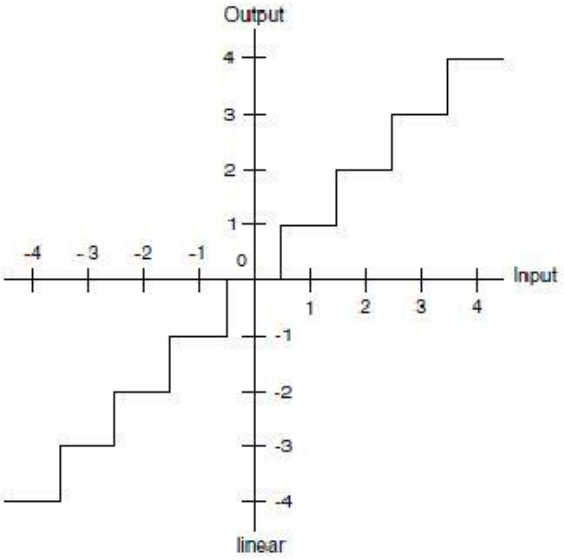

Figura 2.9 Cuantificador uniforme

6. El bloque de los coeficientes ya cuantificados, es reordenado mediante una lectura que puede ser realizada en diferentes direcciones: en diagonal con sentido hacia arriba a la derecha, en vertical o en horizontal. El motivo de esta lectura peculiar es agrupar el mayor número posible de coeficientes no nulos al principio de la trama de bits y obtener secuencias de coeficientes nulos lo más largas posibles, de esta forma los codificadores entrópicos son capaces de lograr una compresión más eficiente.

7. Por último, los coeficientes cuantificados y reordenados, son sometidos a un proceso de codificación entrópica a partir del cual se obtiene la secuencia de bits comprimida que representa la información contenida en el plano que se deseaba codificar, Fn.

En esta etapa, también se codifica información adicional necesaria para realizar el proceso de codificación, como pueden ser las cabeceras o los vectores de movimiento.

Las etapas descritas en los pasos anteriores corresponden al flujo de codificación presente en un codificador de vídeo, pero los codificadores también incluyen un flujo de reconstrucción. Este flujo se incluye debido a que la predicción construida por el codificador debe estar basada en información previamente codificada, obteniendo así la misma información que el decodificador. De no ser así, el decodificador no sería capaz de construir la misma predicción que el codificador, con lo cual, el residuo decodificado no estaría relacionado con dicha predicción, lo que implicaría una acumulación de errores adicionales que harían imposible la correcta decodificación de una secuencia de vídeo.

Las etapas más importantes del flujo de reconstrucción son las siguientes:

1. Al inicio del flujo están presentes los coeficientes transformados y cuantificados, los cuales se corresponden con X en la figura 2.5. A estos coeficientes se les aplica una operación de reescalado, que es el proceso inverso a la cuantificación. Con este proceso, se obtiene una reconstrucción de los coeficientes de la DCT, los cuales no son iguales a los originales debido a que, como se indicó anteriormente, el proceso de cuantificación introduce pérdidas.

2. Los coeficientes reconstruidos son sometidos a un proceso de transformación inversa IDCT (*Inverse Discrete Cosine Transform*), dando lugar a un residuo reconstruido Dn´.

3. El residuo reconstruido Dn' se suma a la predicción P, obteniendo como resultado la reconstrucción de un bloque. Todos los bloques reconstruidos pertenecientes a un mismo plano, formarán un plano reconstruido Fn´. Cabe destacar que este plano reconstruido Fn´ es una versión con pérdidas del plano original a codificar Fn, pero es a su vez la misma reconstrucción que se obtiene en el decodificador siempre que no existan pérdidas durante la transmisión.

2.4.2 Proceso de decodificación

En esta sección se desarrollarán las diferentes etapas que ha de seguir el decodificador para poder obtener la reconstrucción de un plano, a partir de la secuencia de bits generada por el codificador. Se puede apreciar que varias de las etapas presentes en el decodificador están incluidas en el flujo de reconstrucción del codificador, debido a la necesidad de que tanto el codificador como el decodificador, empleen el mismo plano reconstruido.

1. La secuencia de bits generada por el codificador llega al decodificador y atraviesa un proceso de decodificación entrópica con el fin de obtener los coeficientes cuantificados de la DCT y la información adicional correspondientes a cada bloque.

2. Para cada bloque por separado, los coeficientes son reordenados, de tal manera que el resultado sea una matriz de coeficientes DCT cuantificados. Dicha matriz está representada como X en la figura 2.6.

3. A partir de esta etapa, los pasos a seguir son idénticos a los realizados en el flujo de reconstrucción localizado en el codificador, los cuales han sido descritos anteriormente.

   Se aplica un reescalado, a partir del cual se obtiene una matriz de coeficientes reconstruidos, haciendo hincapié en que estos coeficientes serán distintos a los coeficientes originales debido a las pérdidas de información relacionadas con el proceso de cuantificación.

4. Se emplea la IDCT sobre los coeficientes reconstruidos, lo que da lugar al residuo reconstruido Dn'.

5. A partir de la información adicional incluida en la secuencia de bits generada por el codificador, el decodificador puede construir una predicción, identificada como P en la figura 2.6, la cual será igual que la calculada en el codificador.

6. Por último, el residuo reconstruido Dn' se suma a la predicción obtenida en el paso anterior, obteniendo como resultado la reconstrucción de un bloque. Por medio de la reconstrucción de todos los bloques correspondientes a un mismo plano, se obtiene la reconstrucción de dicho plano Fn'.

2.5 Tipos de plano

En función de las predicciones disponibles para cada plano, las secuencias de vídeo comprimido están formadas por tres tipos de plano: Intracodificados (I), Predictivos (P) y Bidireccionales (B).

Los planos I se decodifican por sí mismos, es decir, no necesitan información de otros planos para ser codificados, por lo que sólo emplean la predicción Intra. Permiten corregir errores en la propagación y cuestiones relacionadas con el acceso aleatorio, lo que permite acceder al vídeo decodificado en diferentes instantes de tiempo, esto conlleva a que almacenen una gran cantidad de información respecto a los otros tipos de plano. Debido a esto, no se transmiten de manera muy frecuente, sino que lo hacen de forma periódica.

Los planos P, admiten tanto la predicción Inter como la predicción Intra. La responsabilidad de escoger el mejor tipo de predicción para cada bloque, recae sobre el codificador. En este tipo de planos, el orden de codificación coincide con el orden de visualización, por lo tanto, la predicción Inter se lleva a cabo a partir de planos anteriormente codificados correspondientes al pasado. Este patrón se denomina como patrón IP. Al necesitar de información previa, este tipo de planos no puede ser utilizado para acceder aleatoriamente a una secuencia de vídeo decodificada.

Tanto los planos I como P, puede ser usados como planos de referencia en el proceso de estimación de movimiento. No obstante, si se emplea como referencia un plano P y se produce algún error en la transmisión, dicho error se propagará. En cambio si el plano de referencia es de tipo I, los errores de propagación pueden ser solventados. En la figura 2.10 se muestra un ejemplo de este tipo de patrón.

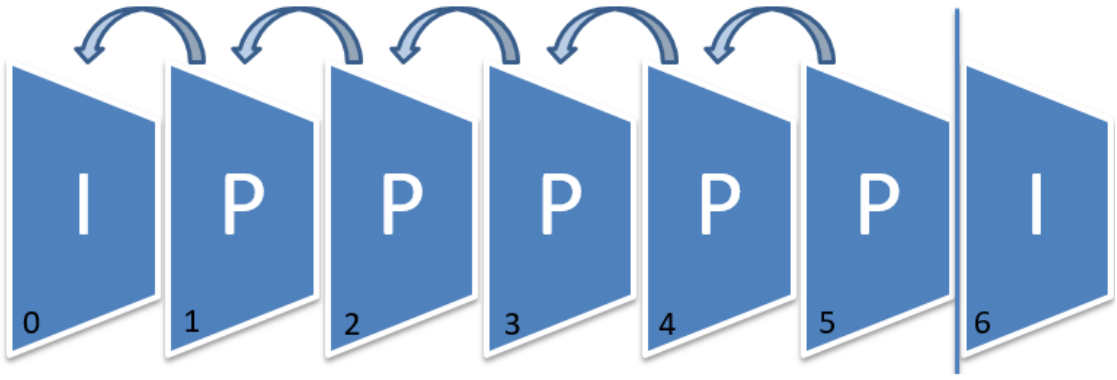

Figura 2.10 Patrón IP

CAPÍTULO 3

ESTÁNDAR DE CODIFICACIÓN HEVC

3. ESTÁNDAR DE CODIFICACIÓN HEVC

3.1 Introducción

El estándar de codificación de vídeo HEVC (*High Efficiency Video Coding*) es el último estándar desarrollado en conjunto por los organismos de estandarización ITU/T *Video Coding Experts Group* e ISO/IEC *Moving Experts Group*. La primera versión de este estándar fue publicada a principios del año 2013, a la que posteriormente se fueron añadiendo diversas mejoras como las extensiones de codificación escalable, multipantalla o aquellas relacionadas con el vídeo en 3D, entre otras.

Debido a la creciente demanda, en los últimos años, de vídeos en alta definición por parte de la televisión, de la red o del amplio abanico de aplicaciones multimedia existentes, y al desarrollo de nuevas resoluciones de vídeo cada vez más grandes, surgió la necesidad de la creación de un nuevo estándar de codificación. El objetivo de HEVC es lograr una capacidad de compresión más eficiente sin que esto suponga un deterioro de la calidad visual.

El estándar HEVC duplica la tasa de compresión respecto a su estándar predecesor, H.264/AVC, lo que significa que para codificar una misma secuencia de vídeo con la misma calidad resultante, HEVC emplea la mitad de datos que los necesarios por parte de H.264/AVC.

HEVC utiliza un modelo de codificación híbrido que incluye nuevas herramientas de codificación que le permiten alcanzar una compresión más eficiente. Este estándar ha sido diseñado para ser compatible con las aplicaciones disponibles de H.264/AVC, y está orientado principalmente en dos aspectos: trabajar con resoluciones de vídeo digital cada vez mayores e incrementar el uso de arquitecturas de procesamiento en paralelo.

El estándar HEVC, no dispone de ninguna técnica específica que consiga por sí sola una relación de compresión tan alta. Esto se consigue gracias a la combinación de pequeñas mejoras aplicadas a las herramientas de codificación ya existentes, en las cuales se centrará este capítulo.

3.2 Diagrama de bloques HEVC

Tal y como se ha mencionado anteriormente, HEVC está basado en un modelo de codificación híbrido similar al descrito en el capítulo 2. En la figura 3.1 se muestra el diagrama de bloques correspondiente al estándar HEVC.

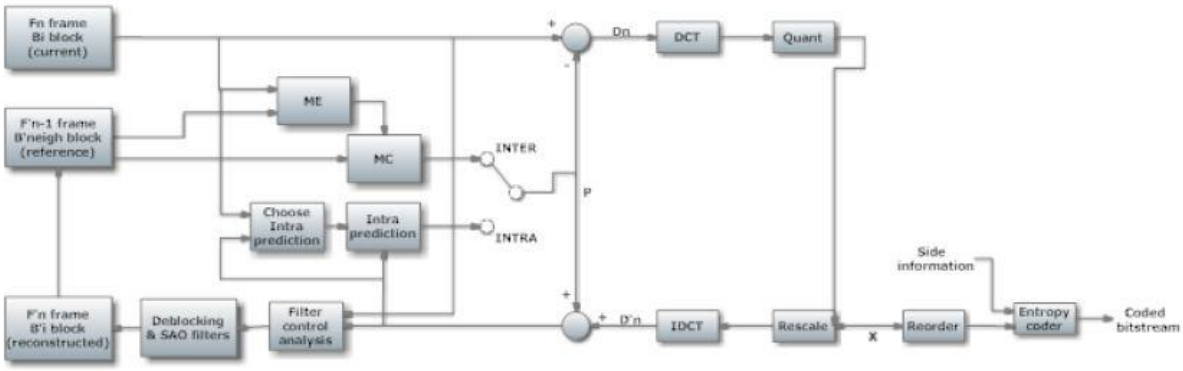

Figura 3.1 Diagrama de bloques de un codificador de vídeo HEVC

Como puede observarse, este diagrama es semejante al esquema de codificación explicado en el capítulo 2, salvo que en el flujo de reconstrucción se introducen dos filtros: *deblocking* y *SAO* cuyo objetivo es reducir la distorsión entre los bordes de los bloques.

El resto de procesos implicados, ya han sido detallados en el capítulo 2.

3.3 Estructura de árbol

Tal y como se ha mencionado anteriormente, la imagen se divide en pequeñas unidades de trabajo de diferentes tamaños denominadas CTBs.

Al llevar a cabo el proceso de codificación, se puede codificar un CTB completo o dividir dicho CTB en unidades más pequeñas, conocidas como *coding unit* (CU). La división se realiza dando lugar a la estructura de un árbol. El codificador es el encargado de determinar el tamaño idóneo de las CU, en función de las características de la zona del plano que representan, aplicando un método de optimización que tiene en cuenta la tasa y la distorsión generada con cada opción de codificación. Dicho

método será explicado con mayor detalle en el apartado 3.9. Esto se traduce en que, para zonas complejas de la imagen, el CTB será dividido en un número elevado de CUs, ya que permitirán predicciones más precisas, mientras que en zonas homogéneas se utilizará un número menor de CUs al no requerir tanta precisión en el cálculo de las predicciones. El tamaño de las CUs varía desde 8 x 8 hasta la dimensión del CTB, dependiendo de la profundidad del árbol en la que esté situada la CU a codificar, es decir, cuanto mayor sea la división efectuada, mayor será la profundidad del árbol y menor será el tamaño de las CUs. En la figura 3.2 se muestra un ejemplo de la división de un CTB aplicando la estructura en forma de árbol.

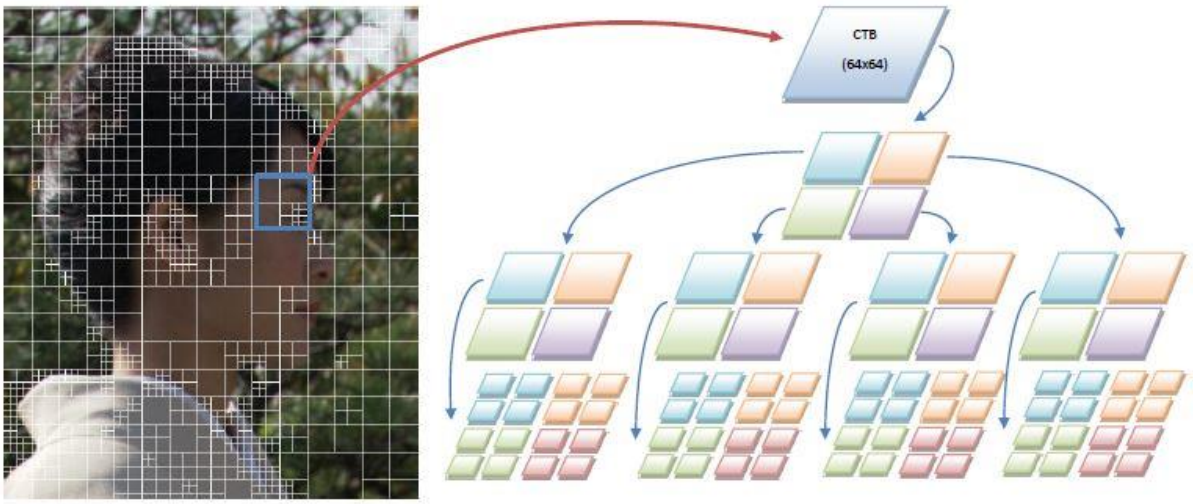

Figura 3.2 División de un CTB aplicando la estructura en forma de árbol

Para cada CU, independientemente de su tamaño, el codificador evaluará diferentes tipos de particiones para calcular las predicciones, denominados *Prediction Units* (PU), que podrán ser del tipo Inter o Intra dependiendo de si se utilizan técnicas de predicción temporales o espaciales, respectivamente.

Respecto a la etapa de transformación, la CU puede considerarse la raíz de otra estructura de árbol, de la cual, derivan las unidades que llevarán a cabo la transformación del residuo resultante de la diferencia entre la imagen original y la predicción. Estas unidades son llamadas *transform units* (TU) y tienen un tamaño que puede variar desde 4 x 4 hasta 32 x 32 píxeles, dependiendo de la profundidad en la que esté situado dicha TU.

3.4 Predicción Intra

Antes de especificar los métodos de predicción Intra incluidos en HEVC, es necesario aclarar la división de las CUs en PUs. Cada CU puede dividirse de dos formas diferentes: la primera, podría no considerarse una división como tal, ya que consiste en que la PU tendrá siempre el mismo tamaño que la CU, siempre que el tamaño de la CU sea superior al tamaño mínimo disponible, 8 x 8 píxeles. La segunda forma, consiste en dividir la CU en cuatro PUs de igual tamaño, lo que sólo se llevará a cabo si el tamaño de la CU a dividir coincide con 8 x 8 píxeles, su tamaño mínimo.

HEVC define diferentes formas de combinar los valores de los píxeles vecinos para formar la predicción Intra, los cuales se presentan a continuación:

1. Intra_Angular. En HEVC se definen un conjunto de hasta 33 direcciones de predicción, mostradas en la figura 3.3. Todas estas direcciones permiten dar una gran cobertura espacial, consiguiendo así predicciones muy precisas.
2. Intra_planar. Con el fin de evitar discontinuidades a lo largo de los bordes de los bloques, este tipo de predicción emplea valores medios de los resultados obtenidos tras aplicar dos predicciones lineales, para obtener la predicción final.
3. Intra_DC. Teniendo también como objetivo reducir las discontinuidades presentes en los bordes, este método usa la media de diversas muestras de píxeles en la vecindad espacial de la CU para construir la predicción.

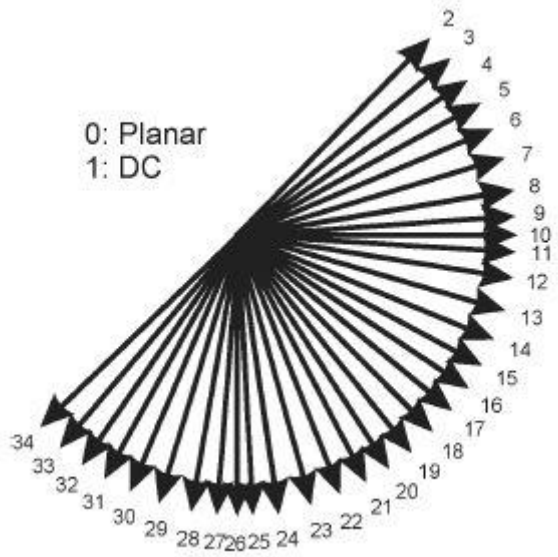

Figura 3.3 Direcciones de predicción para las PUs de tipo Intra

El codificador es el encargado de elegir cuál es el método de predicción Intra que debe aplicar, basándose en el proceso de optimización tasa-distorsión.

3.5 Predicción Inter

3.5.1 Modos de visión

Las diversas formas en las que una CU puede ser dividida, dando lugar a las diferentes PUs, se muestran en la siguiente figura 3.4.

En el modo 2N x 2N no existe división como tal. En el modo N x 2N se realiza una división vertical equitativa, dando lugar a dos PUs verticalmente iguales, mientras que el modo 2N x N es similar al modo anterior pero en este caso la división se realiza de forma horizontal, obteniendo dos PU horizontalmente iguales. El modo N x N sólo puede aplicarse cuando el tamaño de la CU corresponde al valor mínimo definido para este tipo de unidad, dicha CU es dividido equitativamente en horizontal y en vertical, produciendo cuatro PU iguales.

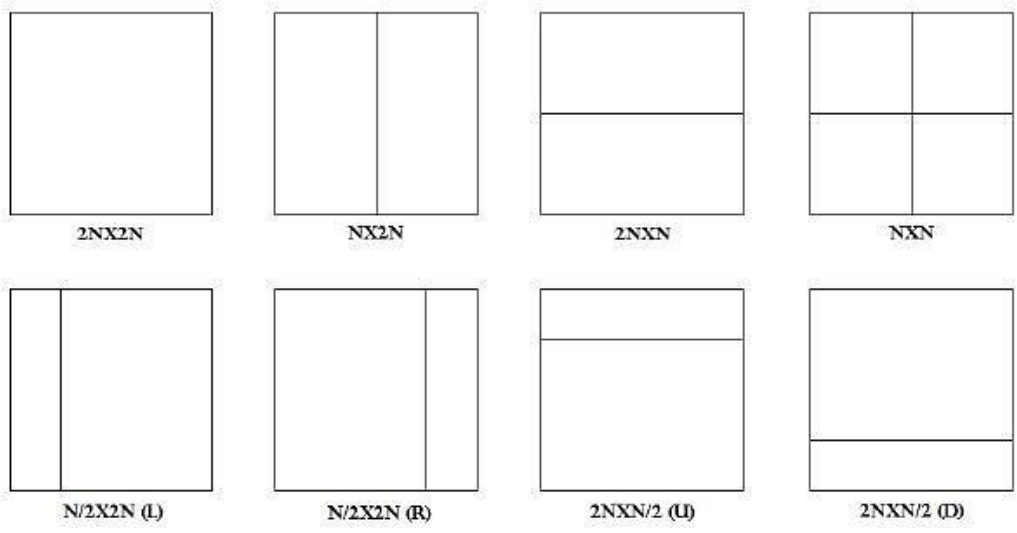

Figura 3.4 Modos de división de una CU en la predicción Inter

Cabe resaltar como novedad, la inclusión en HEVC de los modos N/2 x 2N (*left*), N/2 x 2N (*ritght*), 2N x N/2 (*up*) y 2N x N/2 (*down*), correspondientes a divisiones asimétricas

lo que permite obtener una predicción mucho más exacta. Sin embargo, este tipo de divisiones asimétricas sólo pueden aplicarse si N es mayor o igual a 8.

Toda esta versatilidad de tamaños y divisiones, permite que HEVC pueda realizar el proceso de estimación de movimiento con una precisión muy alta, lo que a su vez, permite lograr una capacidad de compresión mucho más eficiente. No obstante, todas estas herramientas disponibles implican un grado de complejidad computacional muy elevado. De hecho, todos los tipos de predicción disponibles, tanto Inter como Intra, deben ser evaluados para cada CU considerada en el árbol de codificación, lo que se traduce en un tiempo de codificación muy elevado.

3.5.2 Interpolación de muestras fraccionarias

HEVC puede usar vectores de movimiento que apuntan a posiciones entre píxeles, es decir, valores no enteros, con una precisión de un cuarto de píxel. En tal caso, es necesario generar una predicción para las posiciones cuyo valor no sea un número entero, lo que se hace a través de un proceso de interpolación.

Para una muestra de luminancia, este tipo de interpolación emplea un filtro de orden ocho para las posiciones cuya resolución sea la mitad de una muestra, y por otro lado, emplea un filtro de orden siete para las posiciones cuya resolución sea la cuarta parte de una muestra. HEVC es capaz de generar todas las muestras de valor no entero sin aplicar ningún redondeo y junto con el uso de filtros más grandes, mejora la precisión y simplifica la arquitectura de este proceso con respecto a estándares previos.

En la figura 3.5 se muestra un ejemplo de este proceso de interpolación. Las muestras etiquetadas como A_{ij} representan muestras de luminancia situadas en posiciones cuyo valor sea un número entero, mientras que, el resto de etiquetas representan muestras localizadas en posiciones cuya resolución no es un número entero, las cuales han sido generadas mediante interpolación. Las muestras a_{0j}, b_{0j}, c_{0j}, d_{i0}, h_{i0} y n_{i0} han sido calculadas a partir de A_{ij}, empleando el filtro de orden ocho o de orden siete, en función de la resolución de la posición en la que esté situada la muestra correspondiente. Las muestras e_{00}, f_{00}, g_{00}, i_{00}, j_{00}, k_{00}, p_{00}, q_{00} y r_{00} se obtienen a partir de aplicar el filtro correspondiente a las posiciones verticalmente adyacentes a_{0j}, b_{0j} y c_{0j}.

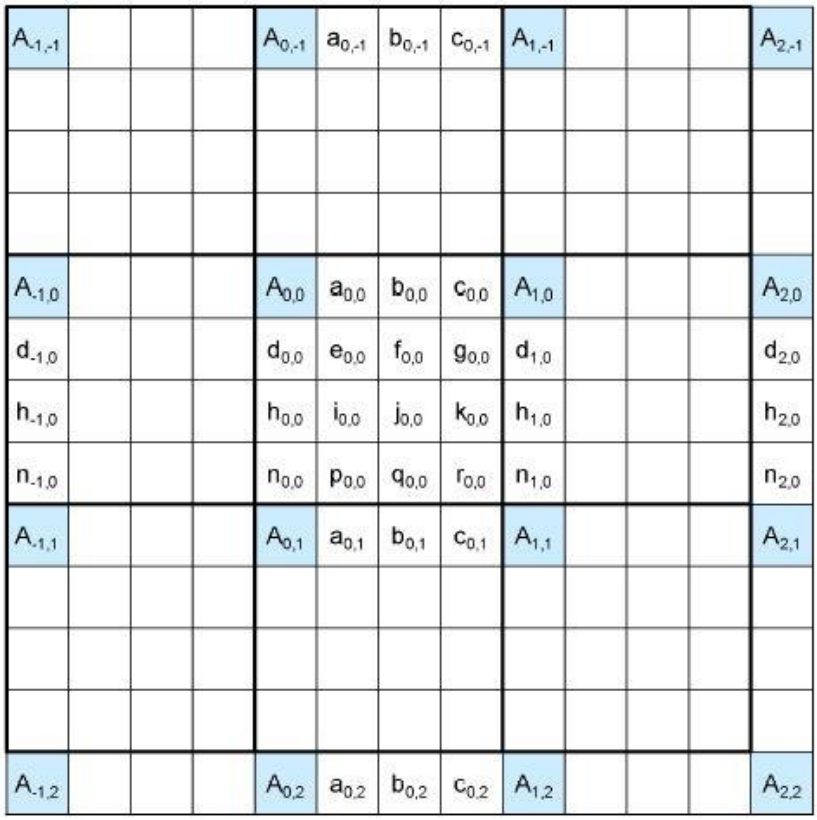

Figura 3.5 Interpolación de muestras cuya resolución es fraccional

3.5.3 Merge mode

HEVC dispone de un método denominado *merge mode*, cuya función principal es reducir el bitrate resultante de codificar una CU, a partir de la información relacionada con el movimiento de los bloques vecinos espaciales y temporales.

El *merge mode* precisa identificar un plano de referencia y un candidato para el vector de movimiento, por lo tanto, necesita incluir cierta información secundaria en el bitstream. La elección del vector de movimiento adecuado entre diversos candidatos es conocido como *motion vector competition scheme*. El conjunto total de vectores de movimiento candidatos está formado por vecinos espaciales, vecinos temporales y otras muestras generadas. En la figura 3.6 pueden apreciarse las distintas posiciones de cinco candidatos espaciales con su orden de disponibilidad correspondiente. Tras validar los candidatos espaciales, se eliminan aquellas posiciones que contengan la misma información que otras.

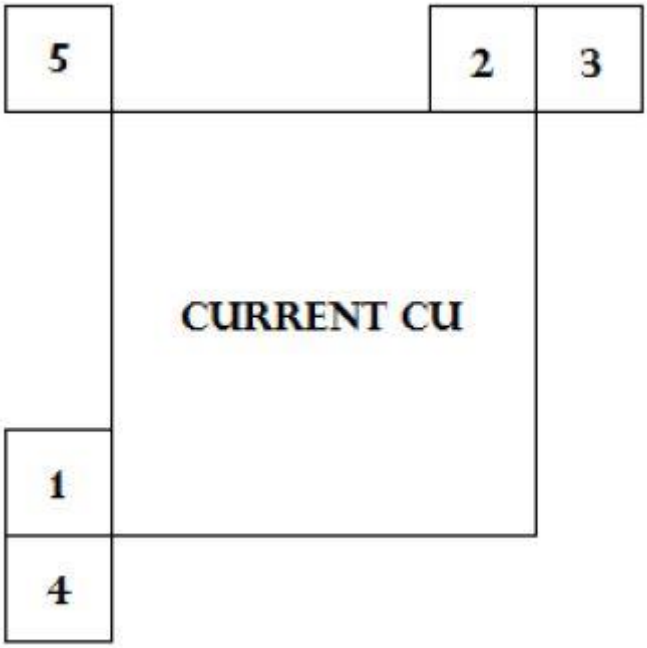

Figura 3.6 Posiciones de candidatos espaciales en el *merge mode*

Para obtener el candidato temporal se empleará como primera opción, la posición que está situada justo por debajo y a la derecha de la PU correspondiente al plano de referencia. En el caso de que esta posición no se encuentre disponible, se utilizará la PU situado en la posición central. Además, HEVC permite seleccionar el plano de referencia a la hora de escoger un candidato temporal para el vector de movimiento, lo cual otorga un alto nivel de flexibilidad.

El número máximo de candidatos se especifica como C, y está incluido en la cabecera de información del *slice*. Si el número total de candidatos espaciales y temporales supera C, únicamente se tendrán en cuenta C – 1 candidatos espaciales. En cambio, si el número de candidatos es menor que C, se generarán candidatos adicionales hasta alcanzar dicho valor.

3.5.4 Predicción del vector de movimiento

La información relacionada con los vectores de movimiento conlleva una gran parte del bitstream generado por el codificador. Teniendo en cuenta que en áreas cercanas es muy posible que el movimiento sea similar, el codificador utilizará la correlación

entre vectores de movimiento de CUs vecinas para construir una predicción del vector correspondiente.

De manera similar al *merge mode*, HEVC permite elegir la predicción del vector de movimiento más óptima entre diversos candidatos. La diferencia entre la predicción y el vector de movimiento actual, así como el índice que identifique el candidato seleccionado, son transmitidos al decodificador.

Se seleccionarán exclusivamente dos candidatos espaciales de los cinco candidatos mostrados en la figura 3.7. El primer candidato será elegido a partir de las posiciones vecinas de la izquierda, numeradas como 1 y 4 en dicha figura, mientras que el segundo candidato procederá de las posiciones vecinas superiores, numeradas como 2, 3 y 5 en la figura 3.7, dependiendo de la disponibilidad de estas posiciones. En el caso de que estén disponibles diversos vecinos, el candidato se seleccionará mediante el orden indicado en tales posiciones.

HEVC sólo permite un número reducido de candidatos a vectores de movimiento con el fin de reducir la complejidad, ya que el proceso de estimación de movimiento es uno de los que más carga computacional requiere.

Si el índice del plano referencia de la PU vecina no es similar al índice de la PU actual, se empleará una versión escalada del vector de movimiento. Esta escala estará relacionada con la distancia temporal entre el plano actual y el plano de referencia indicado en el índice.

Cuando el número de candidatos sea menor que dos, se incluirá un candidato temporal o un vector de movimiento equivalente a cero, lo que garantiza que el número de predicciones siempre será dos, de esta forma basta con codificar un flag que identifique la predicción utilizada, en lugar de codificar ambas predicciones.

3.6 Transformación

Una vez se haya obtenido la predicción de una CU, se calculará el residuo a transformar como la diferencia entre el bloque original y la predicción de ese bloque. Tal y como se ha comentado anteriormente, la CU puede considerarse la raíz de una estructura árbol de la que derivan las unidades que efectuarán el proceso de transformación, las TUs. El residuo puede dividirse de diversas formas, teniendo en cuenta que las dimensiones de las TUs van desde 4 x 4 hasta 32 x 32 píxeles, las cuales

varían en función de la profundidad en la que se encuentre dicha TU. Un ejemplo de este tipo de visión, se muestra en la figura 3.7

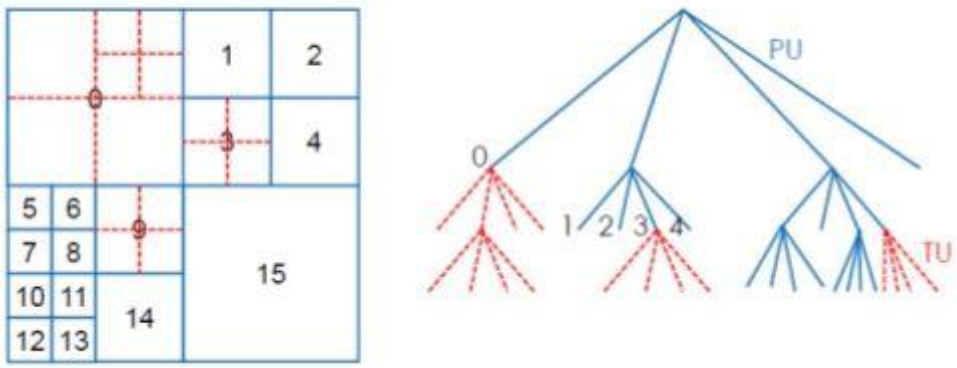

Figura 3.7 Ejemplo de división en PUs y TUs

La transformada empleada en HEVC es la DCT, por medio de la cual, se convierte la matriz del residuo en otra matriz del mismo tamaño que contiene los coeficientes con las amplitudes asociadas a las frecuencias presentes en dicho residuo.

Para llevar a cabo este proceso, en HEVC se aplican dos transformadas, una en dirección horizontal y otra en dirección vertical, dando lugar a una transformada bidimensional. Los coeficientes de la matriz transformada, se obtienen a partir de aplicar una aproximación a escala de las funciones básicas de la DCT, simplificando las operaciones matemáticas a realizar y maximizando la precisión del proceso.

3.7 Cuantificación

Respecto al proceso de cuantificación, cabe recordar que consiste en muestrear la amplitud de los coeficientes obtenidos tras aplicar la DCT, los cuales sólo podrán tomar un valor restringido entre un conjunto de valores limitados.

Matemáticamente, la cuantificación se define como:

$$X_{ij} = round\,(Z_{ij} / Q_{step}),\qquad (3.1)$$

donde Z_{ij} es el coeficiente DCT en la posición ij de la matriz transformada, Q_{step} es el escalón de cuantificación y X_{ij} es la salida del cuantificador, es decir, el valor del coeficiente ya cuantificado en la posición ij. Como puede observarse en la ecuación 3.1, al redondear se generan pérdidas de información, ya que tras efectuar esta operación el valor original no puede ser recuperado.

El Q_{step} es un parámetro muy importante, debido a que determina la relación entre la cantidad de valores de salida, las pérdidas producidas y la cantidad de bits generada. En la figura 2.9 se muestra el ejemplo de un cuantificador uniforme, en ella puede apreciarse que cuanto mayor sea el conjunto de valores de salida disponible, es decir cuanto menor sea el valor del Q_{step}, se producirá menor error ya que el valor cuantificado será más semejante al valor original, aunque será necesario mayor número de bits para poder representar los valores de salida. Cuando Q_{step} aumente, el resultado será el opuesto.

HEVC emplea un esquema de cuantificación URQ (*uniform reconstruction quantization*) controlado por el denominado parámetro de cuantificación (QP). Este parámetro está definido entre 0 y 51 y está relacionado con el Q_{step} tal y como se muestra en algunos ejemplos de la tabla 3.1

Q_{step}	0.62	1.25	2.5	5	10	20	40	80	160	240
QP	0	6	12	18	24	30	36	42	48	51

Tabla 3.1 Relación entre QP y Q_{step}

Aplicar la DCT no comprime los datos, exclusivamente organiza la información con el objetivo de facilitar la compresión. El proceso de cuantificación sí comprime la cantidad de datos procesados, ya que elimina los coeficientes de menor energía a expensas de generar pérdidas asumibles de información.

3.8 Codificación entrópica

La codificación entrópica es un tipo de codificación sin pérdidas de información, que genera el bitstream asociado a los datos obtenidos a través de los procesos explicados previamente.

Para llevar a cabo este proceso, HEVC define exclusivamente un método, denominado codificación aritmética binaria adaptada al contexto (CABAC). En la figura 3.8 se presenta un diagrama de bloques del método mencionado.

Las principales etapas de este método son las siguientes:

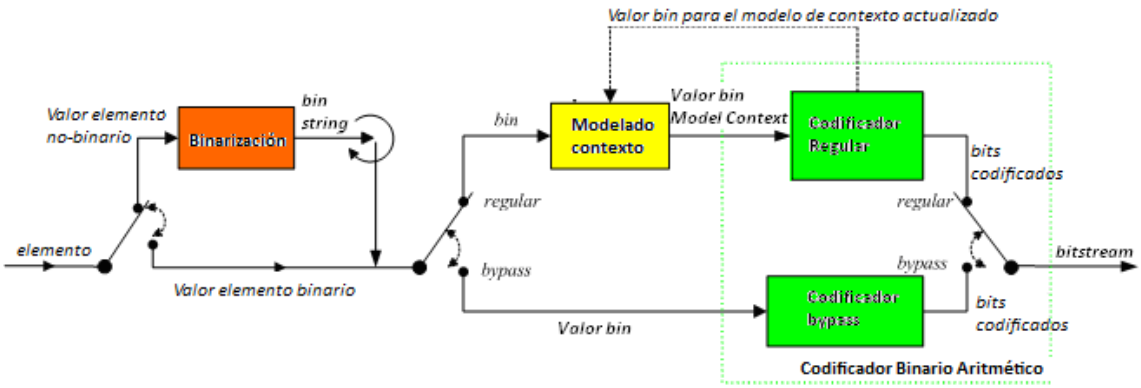

Figura 3.8 Diagrama de CABAC

1. CABAC es un método que divide los datos en diferentes elementos sintácticos binarios, es decir, al inicio del proceso es necesario llevar a cabo una conversión de los elementos a un código binario.
2. Posteriormente se define un modelo de probabilidad para cada uno de los elementos. Este modelo es conocido como contexto, y será elegido a partir de un conjunto de modelos disponibles en función de las estadísticas correspondientes a los símbolos que hayan sido recientemente codificados. Dichos contextos, son modelos de probabilidad asociados a los bits de un símbolo binario, es decir, definen la probabilidad de que cada bit pueda ser 0 ó 1.
3. Tras la elección del modelo de probabilidad, el elemento binario pasa por una etapa de codificación aritmética, dando lugar a la trama de bits que representa la información contenida en el elemento en cuestión.

4. Por último, se actualiza el contexto a partir de los datos estadísticos relacionados con la última codificación realizada.

Hay varias herramientas novedosas definidas en el proceso CABAC de HEVC; por ejemplo, el nivel de profundidad relacionado con las divisiones de las CUs y las TUs es empleado para reducir el número de modelos de probabilidad. Aunque esto parezca contraproducente, HEVC logra una mayor capacidad de compresión gracias a que aprovecha minuciosamente la información relacionada entre los datos que han sido codificados.

Otra herramienta nueva en HEVC, es el escaneo adaptativo de coeficientes, por medio del cual se puede seleccionar la dirección de escaneo más adecuado de los coeficientes DCT cuantificados. Para la predicción Intra, siendo las TUs de tamaño 4 x 4 o de 8 x 8, HEVC permite seleccionar tres direcciones diferentes de escaneo: diagonal, horizontal y vertical, tal y como puede verse en la figura 3.9.

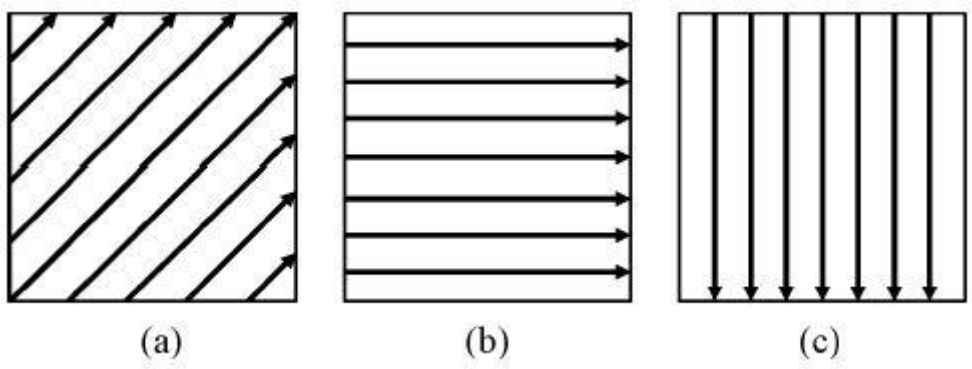

Figura 3.9 Escaneo diagonal (a), horizontal (b) y vertical (c)

Para el resto de tamaños disponibles de una TU, en la predicción Intra, y cuando se emplea la predicción Inter, independientemente del tamaño de la TU, se emplea el escaneo diagonal.

3.9 Optimización tasa-distorsión

Anteriormente se ha mencionado que el estándar HEVC dispone de diversas herramientas para construir la predicción de un bloque que va a ser codificado. Estas herramientas generan diferentes rendimientos en cuanto a calidad resultante y longitud del bitstream asociado. A partir de esta información, el codificador deberá seleccionar la opción que proporcione el rendimiento óptimo a través de un proceso de optimización tasa – distorsión (RDO).

Concretamente, el codificador escogerá aquella herramienta de codificación que minimice la distorsión, entre el bloque reconstruido y el bloque original, y cuya tasa de bits generada sea menor o igual que un valor de tasa máximo permitido. Esto quiere decir que la opción de codificación elegida a la hora de representar la información contenida en un bloque, debe ser aquella que emplee un número de bits menor o igual que la tasa límite establecida y, a su vez, sea capaz de minimizar la distorsión generada.

Esto puede formularse mediante un problema con restricciones, tal y como se muestra en la siguiente ecuación:

$$\min_{\theta} \{ D(\theta) \} \quad sujeto\ a \quad R(\theta) \leq R_c \,, \tag{3.2}$$

donde θ es una combinación de parámetros de codificación, por ejemplo el modo de predicción, el vector de movimiento o el plano de referencia entre otros, $D(\theta)$ representa la distorsión entre el bloque original y el bloque reconstruido, $R(\theta)$ es el número de bits necesarios para codificar el bloque en cuestión, y R_c es la restricción establecida sobre la tasa de bits.

Empleando la formulación de Lagrange, el problema de la ecuación 3.2 puede convertirse en un problema sin restricciones [Everett, 1963]. De esta forma, se obtiene la expresión de un coste J, en el que se pondera la distorsión y la tasa de bits generadas por medio del multiplicador de Lagrange λ. El objetivo será calcular aquel valor de θ que sea capaz de minimizar el siguiente coste J:

$$\min_{\theta} \{ J \} \ donde \quad J(\theta) = D(\theta) + \lambda R(\theta) \,, \tag{3.3}$$

donde el multiplicador de Lagrange λ compara la importancia entre la distorsión y la tasa obtenidas aplicando la opción de codificación θ. Si se resuelve la ecuación 3.3

para un valor específico de λ, se obtiene una solución óptima $\theta^*(\lambda)$ para el problema inicialmente planteado en la ecuación 3.2 para un valor determinado de $R_c = R(\theta^*)$.

Esta solución implicaría evaluar todas las combinaciones posibles de θ para cada bloque y seleccionar el valor óptimo, lo que implicaría un coste computacional excesivo, ya que sería necesario analizar tanto la distorsión como la tasa correspondientes a cada bloque, teniendo en cuenta los diferentes tipos de configuración de los parámetros de codificación. Por ello, en la implementación práctica del estándar HEVC, se llevan a cabo un conjunto de simplificaciones con el fin de facilitar el proceso RDO. Esta serie de simplificaciones, divide el método RDO en dos pasos.

En el primer paso, el codificador debe seleccionar el plano de referencia más adecuado y el vector de movimiento óptimo. Para ello, emplea la siguiente función de coste:

$$J_{motion} = SAD(MV, Ref) + \lambda_{motion} R_{motion}(MV, Ref) \ , \qquad (3.4)$$

donde SAD es la suma del módulo de las diferencias entre el bloque original y el bloque predicho, lo que equivale a una medida de distorsión, λ_{motion} es un multiplicador de Lagrange y R_{motion} es una aproximación del número de bits necesarios para codificar los vectores de movimiento.

Teniendo en cuenta que esta función de coste será evaluada en cada posición relacionada con la estimación de movimiento, por medio de estas simplificaciones, se consigue reducir notablemente la carga computacional del proceso RDO, debido a que permite calcular la distorsión y la tasa de bits de manera simple.

En segundo lugar, cuando el plano de referencia y el vector de movimiento hayan sido seleccionados, el codificador debe elegir el modo de predicción "k" óptimo. Este proceso se denomina *mode decision* (MD) y, en el estándar HEVC, consiste en definir la estructura de codificación de árbol adecuada, especificando los tamaños idóneos para las CUs, PUs y TU. En este caso, la función de coste utilizada es la siguiente:

$$J_{mode,k} = SSD(\{MV\}_k, \{Ref\}_k, k) + \lambda_{mode} R(\{MV\}_k, \{Ref\}_k, k) \ , \qquad (3.5)$$

donde SSD es la suma de las diferencias cuadráticas entre el bloque original y el bloque reconstruido, lo que corresponde a una medida de distorsión, λ_{mode} es un multiplicador de Lagrande diferente al de la ecuación 3.4, y R es la cantidad de bits

necesaria para codificar las cabeceras, los vectores de movimiento, los coeficientes transformados y los índices de los planos de referencia.

Las dos etapas del proceso RDO, están relacionadas mediante los multiplicadores de Lagrange de la siguiente manera [Sullivan and Wiegand, 1998]:

$$\lambda_{motion} = \sqrt{\lambda_{mode}}. \tag{3.6}$$

Por otro lado, para lograr simplificar aún más el proceso y evitar analizar cada uno de los diferentes valores de QP disponibles en el codificador, se definió la relación entre el multiplicador de Lagrange λ_{mode} y la QP como se muestra a continuación:

$$\lambda_{mode} = C \times 2^{\frac{QP}{3}}, \tag{3.7}$$

donde C es una constante que depende del tipo de plano. De esta forma, se pueden obtener ambos multiplicadores de Lagrange λ_{mode} y λ_{motion}, a partir de un valor dado de QP.

Gracias a este proceso, la eficiencia de la codificación mejora significativamente a pesar de que la complejidad del proceso también aumente de forma considerable, debido a la gran diversidad de opciones de codificación disponible que es necesario evaluar. Este proceso es, con diferencia, el que mayor carga computacional requiere, convirtiéndose en el cuello de botella de la codificación de vídeo.

CAPÍTULO 4

ESTADO DEL ARTE

4. ESTADO DEL ARTE

4.1 Introducción

Debido al problema relacionado con la alta carga computacional requerida por el estándar de codificación HEVC, se han desarrollado diversos trabajos orientados a tratar este aspecto.

Tal y como se explicó en el capítulo 3, una CTB puede dar lugar a una estructura jerárquica en forma de árbol, en la que intervienen las CUs, las PUs y las TUs. Por esta razón, podemos encontrar métodos en el estado del arte que agilicen los procesos relacionados con la selección de la partición óptima de cualquiera de estas unidades o que traten el problema como una combinación entre todas ellas.

En este capítulo, se describen brevemente algunas propuestas del estado del arte vinculadas a la reducción de complejidad, las cuales están principalmente orientadas a los métodos relacionados con la decisión del tamaño de las CUs, ya que este trabajo está basado en seleccionar el tamaño y profundidad idóneos a nivel de CU.

4.2 Revisión del estado del arte

En [Jiménez-Moreno et al., 2016], se propone un método de decisión prematura para las CUs basado en un umbral estadístico. A partir del cálculo de las funciones de densidad de probabilidad de los costes RD dado que cierta profundidad del árbol de CUs es óptima, se establece una relación entre la profundidad de las CUs, el contenido de la secuencia del vídeo y el QP, de tal manera que se comprueba que las secuencias de vídeo que contengan zonas homogéneas o movimiento reducido, tienen una probabilidad alta de seleccionar un nivel de profundidad bajo para las CUs, es decir, no resulta necesario dividir la CU ni explorar más niveles de profundidad. Dicho método asegura que los costes RD pueden emplearse como variables de decisión, ya que estos costes dan lugar a pdfs gaussianas que permiten decidir el nivel de profundidad óptimo de forma prematura. Para simplificar el proceso, se establece un umbral asociado a una pdf, de forma que el umbral depende exclusivamente de la media y de la varianza de dicha pdf y de un parámetro relacionado con el objetivo de complejidad deseado, lo que permite al umbral adaptarse a las características de la señal a lo largo del proceso de decisión.

En [Cho et al., 2013], se presenta un algoritmo que combina los métodos de división y de poda, con el fin de establecer el tamaño y la profundidad adecuados para las CUs a lo largo del proceso de codificación Intra. Primero, se aplica el proceso de división mediante el cual, se determina la probabilidad de que la CU sea dividida a partir del análisis de un coste RD de baja complejidad asociado a dicha CU. En el caso de que la CU sea dividida, no es necesario calcular el coste RD total, mientras que, si la CU no es dividida se ejecuta el proceso de poda. Este segundo proceso, permite asegurar que la CU no ha de ser dividida basándose en el estudio del coste RD total perteneciente a la CU en cuestión, en ese caso, el proceso de codificación finalizaría en la profundidad actual de la CU sin considerar las profundidades restantes. Ambos procesos están basados en una regla de decisión bayesiana, de tal forma que los parámetros estadísticos empleados son actualizados de forma periódica con el fin de tener en cuenta las variaciones que puedan producirse en la secuencia de vídeo.

En [Hsu et al., 2013] se plantea un esquema cuyo objetivo es decidir el nivel de profundidad idóneo para las CUs, a partir de la observación de los vecinos espaciales y temporales, el movimiento producido en la secuencia de vídeo y las características relacionadas con la textura de la imagen. El esquema mencionado está dividido en dos partes: decisión a nivel de PU y decisión a nivel de CU. La parte correspondiente al nivel de PU, está diseñada para recudir las operaciones relacionadas con el cálculo de la PU óptima llevadas a cabo en una CU, esto se traduce en que si se cumplen una serie de condiciones para una CU específica, a la hora de calcular la predicción Inter no es necesario comprobar todas las PUs disponibles. Por otro lado, la parte relacionada con el proceso a nivel de CU ha sido desarrollada para evitar que el codificador construya grandes estructuras de árbol, es decir, si una determinada CU cumple con unos requisitos establecidos, los siguientes niveles de profundidad no serán examinados. Estos procesos no se aplican a todos los planos contenidos en un vídeo, se aplican de manera periódica y exclusiva a una serie de planos consecutivos cuya longitud puede variar entre 3, 6, 9, 12 y 15 en función del valor del QP empleado en dicha secuencia.

En [Kim et al., 2013] se propone un método de decisión prematura para el tamaño de las CUs cuyo objetivo es reducir la complejidad del proceso de codificación cuando se emplea la predicción Intra. Se comprobó, por medio de múltiples pruebas en diferentes secuencias de vídeo, que el coste RD asociado a las CUs que no son divididas se concentra principalmente en valores bajos, mientras que, el coste RD asociado a las CUs que sí se dividen en unidades más pequeñas se distribuye normalmente en valores relativamente altos. Por ello, se decidió establecer un umbral de decisión, a partir del cual se decide si seguir analizando los diferentes niveles de profundidad de la CU o si se hace parada prematura y no se evalúan más profundidades. El umbral desempeña

un papel crítico en el método, ya que permite controlar la relación entre la reducción de complejidad y la degradación del proceso de codificación, por esta razón, con el fin de establecer un umbral fiable se define una tasa de error, la cual representa el porcentaje de CUs que han sido codificadas sin dividirse de manera errónea. Si se establece un umbral con un valor alto, el tiempo ahorrado en la codificación será considerable, sin embargo, las pérdidas de calidad visual y la eficiencia de codificación también aumentarán. Por el contrario, si se establece un umbral con un valor bajo, no se conseguirá reducir la complejidad en gran medida y la degradación de la calidad y la eficiencia será ínfima.

En [Zhao et al., 2011] se desarrolla un método que permite reducir el número de candidatos a evaluar en el proceso RDO. Dicho método se basa en modelar la correlación entre la predicción Intra óptima correspondiente a una CU en particular y la predicción Intra de los bloques vecinos, de esta forma, puede reducirse el tiempo destinado a seleccionar el modo de predicción idóneo cuando se emplea la codificación Intra. A partir de estas estimaciones, se comprueba que la dirección de predicción más probable entre bloques vecinos puede ser siempre considerada como un candidato al mejor modo de predicción para la CU actual. El método consta de dos partes, primero se reduce el número de direcciones a tener en cuenta a partir de un proceso RMD (*rough mode decision*), para posteriormente refinar la selección de direcciones candidatas empleando la correlación existente entre la CU actual y los bloques vecinos.

En [Shen et al., 2013] se plantea un método cuya finalidad es reducir el número de niveles de profundidad de las CUs a analizar a lo largo del proceso de codificación. Este método emplea la correlación espacial y temporal para analizar las propiedades de las diferentes regiones que componen los planos de un vídeo. A partir de esta información, se calcula la profundidad óptima para las CUs y se clasifican en diferentes tipos en función de si han sido divididas o no, y del nivel de profundidad asignado. Una vez hayan sido clasificadas las CUs, se aplica un método de decisión que determina si el modo de predicción y los vectores de movimiento pueden representar de manera eficiente el movimiento asociado a las CUs y, de esta forma, eludir el proceso de estimación de movimiento. Se contemplan tres tipos de decisión: la primera se basa en el movimiento homogéneo, la segunda depende del coste RD y la tercera está relacionada con el modo *skip*.

En [Shen et al., 2012] se propone un algoritmo para decidir el tamaño óptimo de las CUs. En primer lugar, se seleccionan aquellas características que puedan ser relevantes a la hora de tomar una decisión correcta, para ello se emplea *Mutual Information*

[Cover and Thomas, 2001]. El objetivo es encontrar las características que consigan maximizar la información mutua, con el fin de lograr una clasificación correcta. En segundo lugar, para minimizar la degradación del proceso de codificación, se establece una regla de decisión bayesiana que se encarga de determinar si las CUs deben dividirse o no. Dicha regla de decisión, se establece a partir de las pdfs bajo cada hipótesis, las cuales son estimadas de forma offline y almacenadas en una tabla de búsqueda. Estos parámetros serán los que determinen el umbral de decisión.

En [Grellert et al., 2013] se presenta un esquema que gestiona la carga de trabajo con el propósito de controlar dinámicamente la complejidad computacional del proceso de codificación. El esquema consiste en un bucle que va analizando los diferentes planos, se calcula el nivel de cómputo alcanzado en un plano y, en función de este valor, se asignan los recursos disponibles para codificar el siguiente plano. Este proceso se realiza de manera continua para todos los planos, lo que permite actualizar de manera dinámica los recursos disponibles para cada plano. Debido a que unas zonas del plano son más complejas que otras, los recursos asignados no pueden distribuirse equitativamente a lo largo de las CUs que componen dicho plano. Por esta razón, también se propone en este trabajo un método de clasificación, el cual está basado en información que ya ha sido codificada con anterioridad; por ejemplo, el nivel de profundidad máximo y la información relacionada con el movimiento de cada CU. Dicho método diferencia tres tipos de CUs en relación a la carga computacional que requiere su codificación: carga baja, carga media y carga alta.

En [Correa et al., 2013] se propone un método de reducción de complejidad basado en la observación de que zonas específicas de un plano con características similares como el movimiento o la textura, emplean la misma profundidad de CUs. Este método estima el nivel de profundidad máximo asociado a una CU a partir de la información aportada por las CUs vecinas espaciales y temporales, con el propósito de evitar examinar el resto de profundidades. Esta estimación se realiza sólo en algunos planos denominados “sin restricciones” destinados a obtener esta información. En el resto de plano, llamados “con restricciones” sólo se podrán evaluar las profundidades anteriormente calculadas.

En [Zhang et al., 2013] se presenta un algoritmo de decisión para la profundidad de una CU que pretende controlar la complejidad en función de la información relacionada con los vecinos espaciales y temporales. Establece un rango de búsqueda adaptativo para las profundidades de cada CU, en función del nivel de profundidad que haya sido más utilizado por sus vecinos.

CAPÍTULO 5

MÉTODO PROPUESTO

5. MÉTODO PROPUESTO

5.1 Introducción

Como se ha visto en el capítulo 3, el estándar HEVC dispone de una variedad de herramientas que permiten lograr una mayor eficiencia en la codificación de vídeo. Sin embargo, el uso de estas herramientas implica un aumento considerable de la complejidad del proceso, y un incremento del tiempo empleado en la codificación.

Específicamente, la herramienta más importante de HEVC es la estructura jerárquica en forma de árbol. Esta herramienta permite dividir los planos en CTBs, los cuales a su vez, pueden también dividirse en otras unidades de diferentes tamaños, las CUs. Cada CU se divide en las PUs, relacionadas con los modos de predicción, y en las TUs, asociadas al proceso de transformación, y el codificador deberá evaluar todas las combinaciones posibles a través del proceso RDO para seleccionar la representación óptima de ese bloque CTB. Por esta razón, dicha herramienta requiere un alto nivel de carga computacional.

En este trabajo, se propone un algoritmo efectivo basado en una decisión prematura del nivel de profundidad asociado a las CUs, cuyo objetivo es reducir el tiempo destinado a codificar una secuencia de vídeo, con el fin de simplificar la complejidad del proceso de codificación. El algoritmo está basado en un enfoque jerárquico, en el cual un test de hipótesis es el encargado de decidir, para cada nivel de profundidad, si es necesario dividir la CU y continuar con el proceso de codificación en el nivel de profundidad posterior o, en su defecto, no aplicar dicha división y finalizar el proceso de codificación para esa CU.

Las principales características de este método residen en dividir cada plano en diferentes regiones, para poder aprovechar la correlación existente entre las diversas zonas presentes en un plano, y calcular los parámetros estadísticos que definen el test de hipótesis de manera online y de forma independiente para cada región considerada. De esta forma se pretende que el sistema propuesto pueda tener en cuenta las características variadas que puedan aparecen en diferentes zonas de las imágenes y, además, adaptarse al contenido de cada vídeo típicamente variable.

En este capítulo se proporciona una explicación detallada del algoritmo propuesto, haciendo hincapié en las propiedades más relevantes.

5.2 Test de hipótesis

El algoritmo se basa en la aplicación de un test de hipótesis binario cuyo umbral está determinado por las características de las diferentes regiones en los planos que componen una secuencia de vídeo. Dicho umbral se actualiza a lo largo del proceso de codificación, es decir, de manera online, lo que significa que el valor del umbral variará en función del contenido del vídeo. Esto permite establecer un umbral fiable para cada situación que se dé a lo largo de la codificación, lo cual ayudará a tomar un mayor número de decisiones correctas, gracias a la adaptación a las propiedades específicas de cada vídeo.

En concreto, nuestra propuesta se centra en la decisión a nivel de CU, ya que ésta es una de las que suponen mayor coste computacional en el diseño de HEVC. Para ser más exactos, el proceso que sigue el codificador a nivel de CU es el siguiente: calcula los costes en cada una de las diferentes profundidades para una CU. Posteriormente compara dichos costes y escoge el nivel de profundidad cuya codificación suponga el menor coste. El número de profundidades que puede emplear una CU, es cuatro, enumeradas por el estándar de 0 hasta 3, de tal forma que la profundidad 0 corresponde a una CU de tamaño 64 x 64, la profundidad 1 corresponde a 4 CUs de tamaño 32 x 32 cada una, la profundidad 2 corresponde a 16 CUs de tamaño 16 x 16 cada una y, por último, la profundidad 3 corresponde a 64 CUs de tamaño 8 x 8 cada una. Como podemos observar, para elegir la profundidad óptima de una CU, el codificador debe analizar de forma previa y exhaustiva cada una de las ramas del árbol generado. Teniendo en cuenta que cada CTB presente en un plano da lugar a una estructura de árbol distinta, la complejidad y la duración del proceso es realmente considerable, por lo que decidimos centrar nuestro trabajo en este proceso que selecciona el tamaño de CU óptimo.

El test de hipótesis podrá seleccionar entre efectuar la división de una CU y analizar el siguiente nivel de profundidad o, por el contrario, no ejecutar la división de dicha CU y excluir el estudio de los niveles de profundidad posteriores. Esto permite reducir considerablemente el número de procesos llevados a cabo durante la codificación y, a su vez, disminuye el tiempo necesario para codificar una secuencia de video. Además, se considerarán diferentes regiones independientes en cada plano del vídeo codificado, definiendo un test de hipótesis para cada una, utilizando parámetros estadísticos independientes para cada región. En la figura 5.1 se muestra el diagrama de flujo que contiene los principales pasos ejecutados en este método.

Siendo más explícitos, la formulación general matemática en la que se basa el test de hipótesis procede de la teoría Bayesiana de la decisión, en la cual se determina que la decisión óptima es aquella que proporciona el menor coste medio, tal y como se establece en la siguiente expresión:

$$D_{j*} = \arg\min_{j} \sum_{i} C_{ji}\, Pr_x(x|H_i)\, Pr(H_i)\,, \tag{5.1}$$

donde D_{j*} representa la decisión óptima, siendo la decisión que minimiza la suma, teniendo en cuenta todas las i hipótesis posibles, de los costes C_{ji} asociados a decidir j cuando la hipótesis correcta es i. Este valor se pondera con las funciones densidad de probabilidad condicionadas de obtener la observación x dada la hipótesis H_i $(Pr_x(x|H_i))$, y con las probabilidades a priori correspondientes a cada hipótesis $(Pr(H_i))$.

En base a la formulación anterior, se puede establecer un test de hipótesis binario compuesto por dos hipótesis, H_0 y H_1, y dos decisiones, D_0 y D_1. Siendo, H_0 la hipótesis asociada a no efectuar la división de la CU y parar el proceso de codificación para dicha CU, H_1 la hipótesis asociada a ejecutar la división de la CU y continuar con el proceso habitual, D_0 la decisión relacionada con la elección de la hipótesis H_0, y D_1 la decisión relacionada con la elección de la hipótesis H_1. Para este caso particular, los costes medios asociados a decidir D_0 o D_1 se definen de la siguiente forma:

$$\overline{C_0}(x) = C_{00}\, Pr_x(x|H_0)\, Pr(H_0) \;+\; C_{01}\, Pr_x(x|H_1)\, Pr(H_1)\,, \tag{5.2}$$

$$\overline{C_1}(x) = C_{10}\, Pr_x(x|H_0)\, Pr(H_0) \;+\; C_{11}\, Pr_x(x|H_1)\, Pr(H_1)\,. \tag{5.3}$$

A partir de las expresiones anteriores puede definirse para este problema el test del cociente de verosimilitudes, también conocido como LRT (*Likelihood Ratio Test*), cuya expresión es la que se muestra a continuación:

$$\frac{Pr_x(x|H_1)}{Pr_x(x|H_0)} \underset{D_0}{\overset{D_1}{\gtrless}} \frac{(C_{10} - C_{00})}{(C_{01} - C_{11})} \frac{Pr(H_0)}{Pr(H_1)}\,. \tag{5.4}$$

En las siguientes secciones de este capítulo, se explicará en detalle los parámetros que componen dicho test.

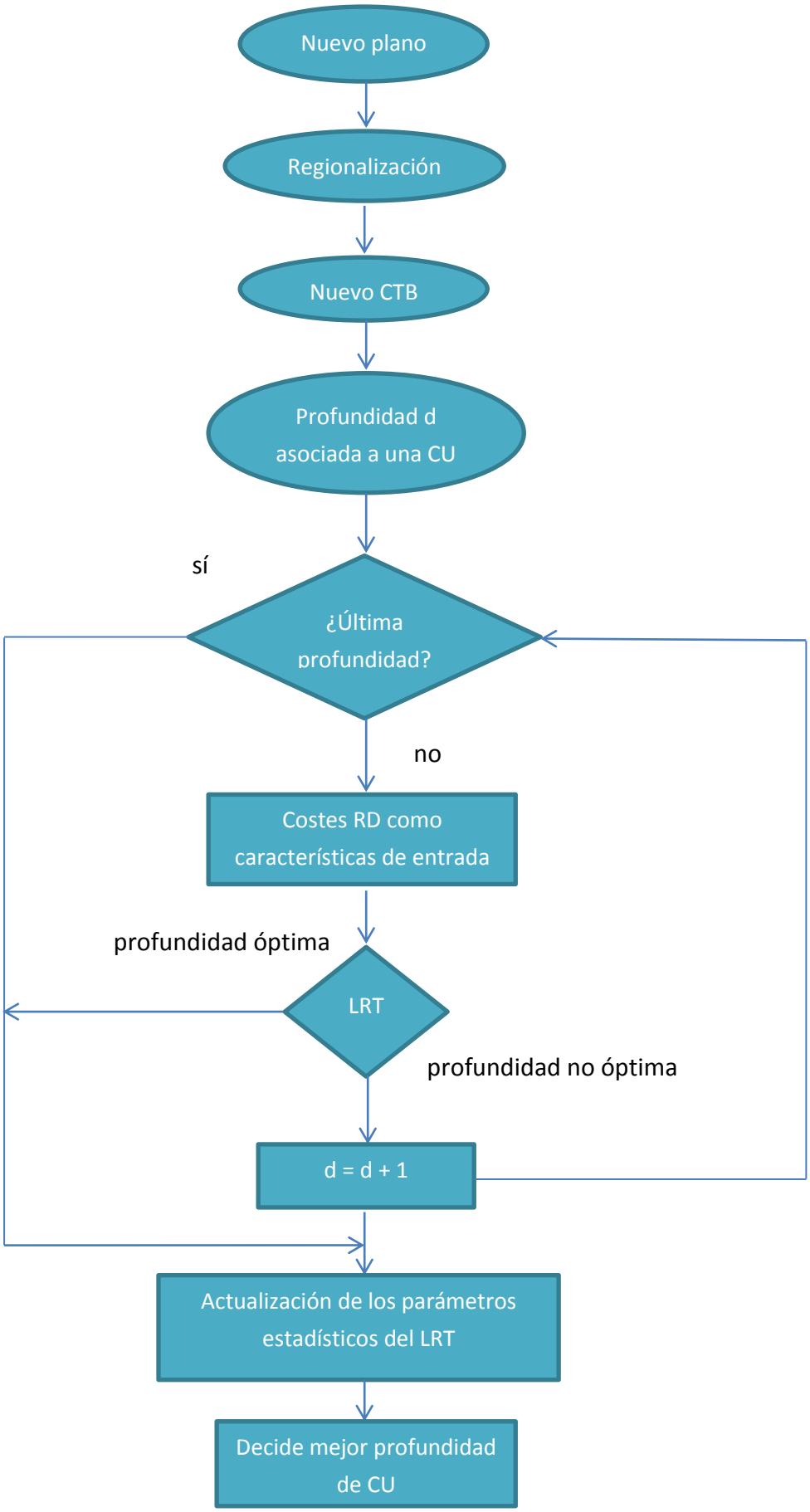

Figura 5.1 Diagrama de flujo

5.3 Variables de entrada

En la sección 3.9, se indicó que el codificador seleccionaba, entre otras características, el nivel de profundidad óptimo de una CU basándose en los costes RD, especificados en la ecuación 3.5, asociados a cada nivel de profundidad de la CU correspondiente.

Para poder aplicar el LRT, definido en la expresión 5.4, el primer paso consiste en seleccionar la característica de entrada x empleada. El objetivo es seleccionar la característica de mayor relevancia, lo que equivale a seleccionar la característica que proporcione pdfs lo más separadas posibles, con lo que al aplicar el LRT con ellas se pueda alcanzar la mayor precisión posible en el proceso de decisión.

Debido a que la elección de la profundidad óptima de una CU, está basada en la evaluación de los costes RD, en este algoritmo se emplean como características de entrada x dichos costes asociados a cada nivel de profundidad. En este caso, se debe evaluar su capacidad para efectuar la decisión prematura de la profundidad óptima. Para ello, se definen las siguientes pdfs:

$$\Pr\left(J_{PU=a^*,depth=d} \middle| depth^* = d\right) , \tag{5.5}$$

$$\Pr\left(J_{PU=a^*,depth=d} \middle| depth^* > d\right) \tag{5.6}$$

donde $J_{PU=a^*,\ depth=d}$ es el coste RD correspondiente a codificar la CU con el modo óptimo de predicción a^* y con una profundidad d condicionado a que dicha profundidad sea la óptima, ecuación 5.5, o que dicha profundidad no sea la idónea, ecuación 5.6. Se selecciona el coste asociado al modo de PU óptimo, dejando que el codificador evalúe todas las PUs disponibles para cada CU que decida comprobar, escogiendo entre ellas la mejor. Seleccionando este coste, se elude entrar en tomar decisiones a nivel de PU permitiendo que sea el codificador el que las haga, centrando el diseño propuesto únicamente a nivel de CU.

En la figuras 5.2 y 5.3 pueden apreciarse varios ejemplos de estas pdfs.

Con el fin de aportar información más precisa, en la tabla 5.1 se muestran diferentes resultados correspondientes a las profundidades 0, 1 y 2 para la secuencia *Blowing Bubbles* teniendo en cuenta diferentes valores de QP. Los parámetros $\mu \mid H_i$ y $\sigma \mid H_i$ indican la media y la desviación estándar, respectivamente, asociadas a la elección de la hipótesis H_i, lo que permitirá comprobar si las estadísticas son suficientemente diferentes para cada hipótesis considerada.

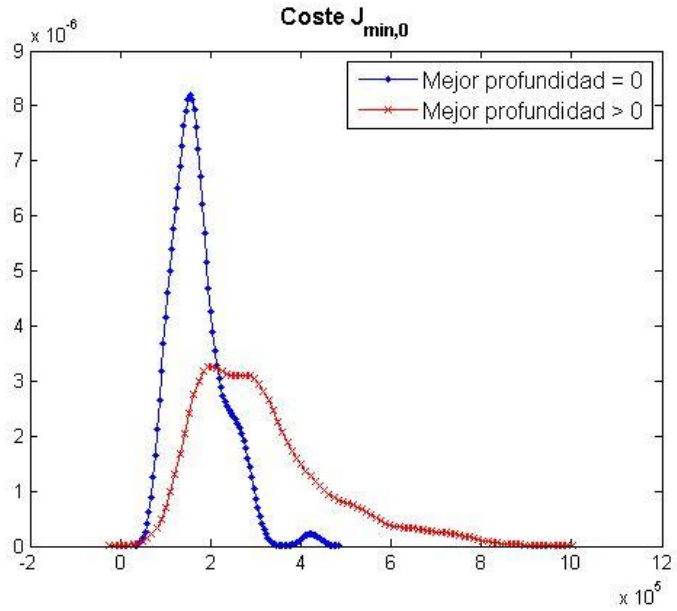

Figura 5.2 Pdf del coste J_{min} asociado a la profundidad 0 para la secuencia *Blowing Bubbles* con QP 32

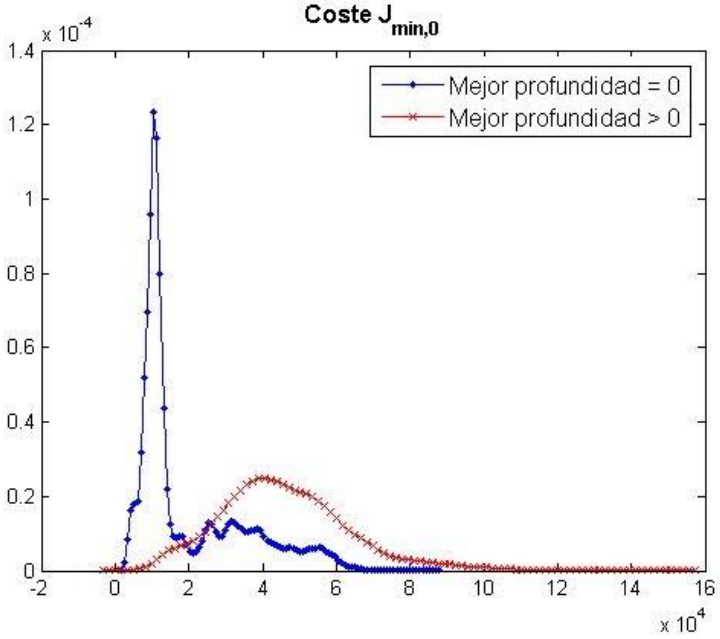

Figura 5.3 Pdf del coste J_{min} asociado a la profundidad 0 para la secuencia *Johnny* con QP 27

QP	Parámetros estadísticos	Profundidad 0	Profundidad 1	Profundidad 2
22	$\mu \mid H_0$	36339	12033	3766
	$\sigma \mid H_0$	10730	4440	1902
	$\mu \mid H_1$	82774	19894	4831
	$\sigma \mid H_1$	29363	8572	2720
27	$\mu \mid H_0$	82706	24428	8081
	$\sigma \mid H_0$	13115	10427	5401
	$\mu \mid H_1$	166857	42743	8997
	$\sigma \mid H_1$	72945	22931	6721
32	$\mu \mid H_0$	171784	50579	15678
	$\sigma \mid H_0$	57678	25921	13367
	$\mu \mid H_1$	312409	79947	17897
	$\sigma \mid H_1$	143560	48376	12945
37	$\mu \mid H_0$	366032	108479	28544
	$\sigma \mid H_0$	139335	66734	28536
	$\mu \mid H_1$	579139	129215	37027
	$\sigma \mid H_1$	255024	81072	22915

Tabla 5.1 Medias y desviaciones estándar de los costes RD al seleccionar la hipótesis H_i para la secuencia *Blowing Bubbles*

A partir de la observación de los resultados de la tabla y las figuras anteriores, se comprueba que las pdfs son lo suficientemente separables, para poder construir el LRT con ellas y determinar la profundidad óptima de una CU de manera fiable. Además, se asume gaussianidad para ambas distribuciones. La razón de asumir esta característica, es que simplifica el proceso de estimación de las pdfs, ya que la expresión del test de hipótesis definido en la expresión 5.4 puede reescribirse de la siguiente manera:

$$\frac{\exp\left(\frac{-\left(J_{PU=a,\ depth=d}-\hat{\mu}_1\right)^2}{2\hat{\sigma}_1{}^2}\right)}{\exp\left(\frac{-\left(J_{PU=a,\ depth=d}-\hat{\mu}_0\right)^2}{2\hat{\sigma}_0{}^2}\right)}\frac{\hat{\sigma}_0{}^2}{\hat{\sigma}_1{}^2} \gtrless \begin{matrix} D_1 \\ D_0 \end{matrix} \frac{\hat{P}(H_0)}{\hat{P}(H_1)}\frac{C_{10}}{C_{01}} \quad , \tag{5.7}$$

donde ,a la izquierda de la expresión, los parámetros $\hat{\mu}_0$ y $\hat{\mu}_1$ representan las medias estimadas de las pdfs condicionadas a las hipótesis H_0 y H_1 respectivamente, y los parámetros $\hat{\sigma}_0{}^2$ y $\hat{\sigma}_1{}^2$ corresponden a las varianzas estimadas de las mismas pdfs. A la derecha de la expresión, $\hat{P}(H_0)$ y $\hat{P}(H_1)$ son las estimaciones de las probabilidades a priori de cada hipótesis, mientras que los costes C_{10} y C_{01} son los costes asociados a las decisiones erróneas, teniendo en cuenta que, en esta expresión, los costes C_{00} y C_{11} toman un valor nulo.

Como puede observarse en la expresión anterior, a partir de la estimación exclusiva de la media y la varianza correspondientes, dichas distribuciones pueden ser definidas. De esta manera, la asunción de gaussianidad simplifica en gran medida nuestro diseño, ya que almacenando un conjunto reducido de valores podremos aplicar el LRT.

En la siguiente sección, se detallará el proceso de estimación de los parámetros que definen dicha expresión.

5.4 Estimación de los parámetros estadísticos

Tomando logaritmos en la expresión 5.7 obtenemos la regla de decisión definitiva empleada en el algoritmo:

$$-\frac{\left(J_{PU=a,\ depth=d}-\hat{\mu}_1\right)^2}{2\hat{\sigma}_1{}^2}+\frac{\left(J_{PU=a,\ depth=d}-\hat{\mu}_0\right)^2}{2\hat{\sigma}_0{}^2}+\ln\frac{\hat{\sigma}_0{}^2}{\hat{\sigma}_1{}^2}\ \gtrless_{D_0}^{D_1}\ \ln\frac{\hat{P}(H_0)}{\hat{P}(H_1)}\frac{C_{10}}{C_{01}}. \qquad (5.8)$$

Cabe resaltar que los valores de $\hat{\mu}_0, \hat{\mu}_1, \hat{\sigma}_0{}^2, \hat{\sigma}_1{}^2$ y las probabilidades a priori, $\hat{P}(H_0)$ y $\hat{P}(H_1)$, son estimados online, es decir, durante el proceso de codificación. Gracias a esto, los parámetros y la decisión del test, se ajustan en función de las diversas características correspondientes al contenido variable de cada secuencia de vídeo, lo que aporta mayor fiabilidad al algoritmo propuesto. Teniendo en cuenta la enorme variabilidad presente en las secuencias de vídeo, este tipo de estimación online supone una gran ventaja con respecto a entrenamientos offline donde la capacidad de generalización se puede ver comprometida debido a esa variabilidad tan pronunciada en este tipo de medio visual

Para llevar a cabo las estimaciones online, se han estudiado dos tipos de métodos: estimación exponencial y estimación aritmética. En la figura 5.4 se muestra un ejemplo del comportamiento de ambos métodos de estimación, en concreto, se muestra el parámetro $\hat{\mu}_0$, la estimación de la media de la pdf $\Pr\left(J_{PU=a,\ depth=d} \middle| depth^* = d\right)$, calculado con los dos tipos de métodos mencionados. En dicha figura, también se incluye los valores medios esperados correspondientes a grupos de 50 muestras, de esta manera se puede comprobar la capacidad de seguimiento de cada uno de los diferentes métodos de estimación.

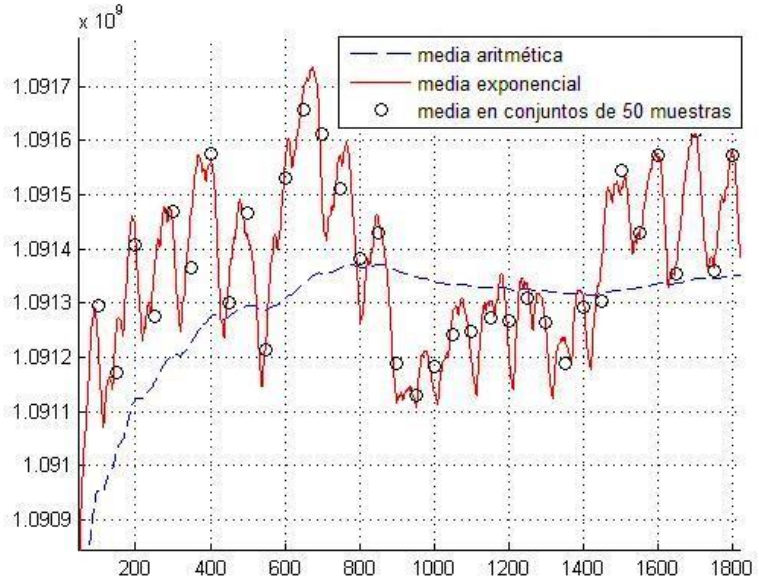

Figura 5.4 Estimación de $\hat{\mu}_0$ empleando los métodos aritmético y exponencial

Como puede observarse en la figura 5.4, la estimación exponencial realiza un mejor seguimiento que la estimación aritmética, ya que los valores estimados se aproximan a los valores esperados gracias a que otorga menor relevancia a las muestras antiguas que a las muestras actuales. En cambio, la estimación aritmética tiene como resultado una capacidad de seguimiento deficiente, a medida que el número de muestras se incrementa.

Por esta razón, para estimar los valores de las medias y las varianzas se emplea un método de estimación exponencial, que se formula como sigue:

$$\hat{\mu}_i(n) = \alpha\hat{\mu}_i(n-1) + (1-\alpha)\cdot J_{PU=a^*,depth=d}(n),\ i=\{0,1\}, \quad (5.9)$$

$$\hat{\sigma}_i{}^2(n) = \beta\hat{\sigma}_i{}^2(n-1) + (1-\beta)(J_{PU=a^*,depth=d}(n) - \hat{\mu}_i(n))^2,\ i=\{0,1\}, \quad (5.10)$$

donde n es el número de veces en el que la hipótesis H_i ha sido seleccionada, $\hat{\mu}_i(n)$ y $\hat{\sigma}_i{}^2(n)$ son los valores de la media y la varianza, respectivamente, estimados en el instante n; $\hat{\mu}_i(n-1)$ y $\hat{\sigma}_i{}^2(n-1)$ son los valores de la media y la varianza, respectivamente, estimados en el instante $(n-1)$; $J_{PU=a^*,depth=d}(n)$ es el coste RD utilizado como característica de entrada al LRT en el instante n. Los parámetros α y β son los factores de olvido del promediado exponencial y determinan la importancia de las muestras antiguas, en comparación con el peso dado a las muestras recientes,

involucradas en el proceso de la estimación. Experimentalmente, se decide que ambos parámetros tomen un valor de 0.95.

En relación a las probabilidades a priori, $\hat{P}(H_0)$ y $\hat{P}(H_1)$, son estimadas a partir del número de veces que ha sido elegida cada hipótesis. Además, con el fin de evitar que una de las hipótesis sea siempre seleccionada, provocando así una relación muy desequilibrada, se limita el valor máximo de estas probabilidades.

Por último, cabe destacar que el test de hipótesis no se aplica hasta que no se haya realizado una estimación razonable de todos los parámetros involucrados, es decir, se establece un número mínimo de muestras a obtener para realizar estimación antes de aplicar el LRT.

5.5 Regionalización

Otra de las principales propuestas de este diseño, junto con la adaptabilidad explicada en el apartado previo, es la regionalización. Considerando que habitualmente existen regiones, en los planos que forman una secuencia de vídeo, que presentan características claramente distintas, es recomendable considerar que las estadísticas asociadas a cada una de estas regiones deben ser independientes, ya que pueden provenir de zonas de la imagen con contenidos muy distintos. Por ejemplo, consideremos el plano mostrado en la figura 5.5, como se puede observar existen claramente zonas diferentes, especialmente aquellas que pertenecen al fondo y las que pertenecen a los objetos en primer plano. Su contenido y, por tanto, sus características son muy distintas, lo que llevarán a que el codificador utilice herramientas diferentes para codificar cada región. Por todo ello, optamos por llevar a cabo una regionalización.

Por todo ello, las zonas correspondientes a una determinada región del cuadro frecuentemente tendrán valores similares de los costes RD, mientras que regiones distintas tenderán a obtener diferentes costes. Por lo tanto, manteniendo independencia estadística entre las regiones consideradas, se evita mezclar datos de características muy diferentes, lo que permitirá obtener distribuciones más separadas.

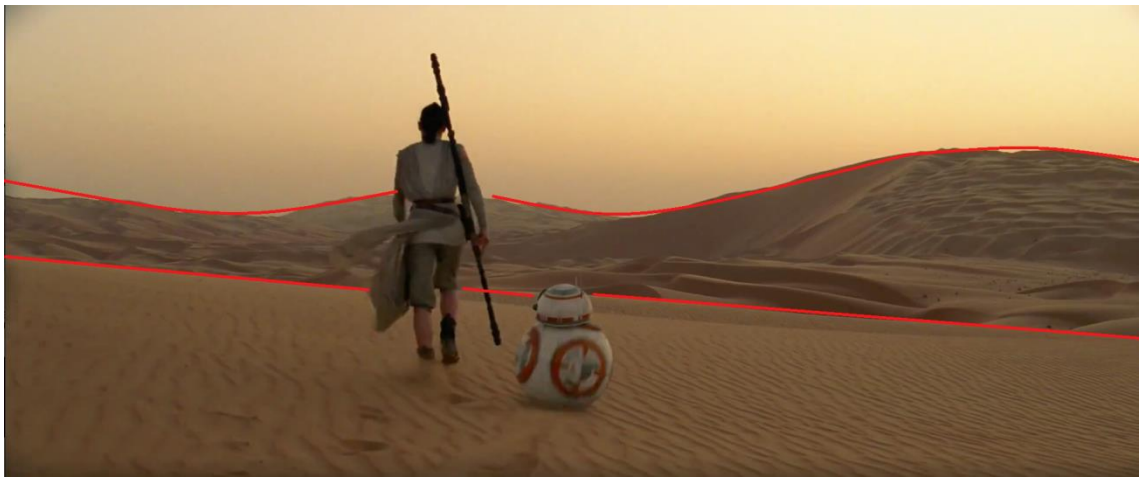

Figura 5.5 Zonas claramente diferenciadas en un plano

A lo largo del proceso de evaluación del algoritmo, también se estudiaron diferentes tipos de regionalización para un cuadro: la división en cruz, es decir, dividir el cuadro por la mitad tanto a nivel vertical como a nivel horizontal, la división en tres regiones horizontales, la división en cuatro regiones horizontales y, aunque técnicamente no es un tipo de regionalización, se consideró la idea de trabajar sin regiones. En la figura 5.6, se muestran los diferentes tipos de divisiones mencionados.

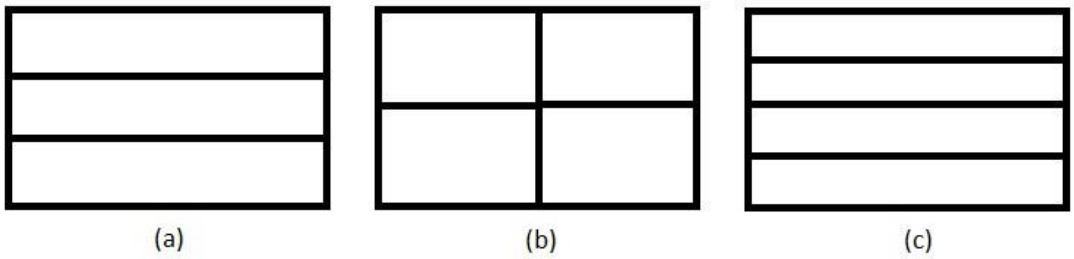

Figura 5.6 División en 3 regiones horizontales (a), en cruz (b) y en 4 regiones horizontales (c)

Se han implementado en total cuatro diseños, uno para cada tipo de división explicado, los cuales se han utilizado para codificar un conjunto de secuencias y medir sus prestaciones en cuanto a ahorro de tiempo (TS) e incremento de bitrate (ΔBR), ambas medidas están definidas en el próximo capítulo. En la tabla 5.2, se muestran un

conjunto reducido de los resultados obtenidos, ya que en el capítulo siguiente se mostrarán los resultados completos.

Secuencia	Sin regiones		Cruz		3 regiones horizontales		4 regiones horizontales	
	ΔBR	**TS**	**ΔBR**	**TS**	**ΔBR**	**TS**	**ΔBR**	**TS**
BasketballDrillText 832x480	7.78	41.31	8.21	41.46	7.33	41.34	7.08	40.80
BQSquare 416x240	1.61	4.66	0.60	3.90	0.61	4.53	0.71	5.45
Cactus 1920x1080	2.79	35.29	3.00	36.08	2.74	36.50	3.19	39.06

Tabla 5.2 Conjunto reducido de resultados correspondientes a emplear diferentes tipos de regionalización

Con los resultados obtenidos, se puede concluir que dividir el plano en tres o cuatro regiones horizontales proporciona mejores resultados en ahorro de tiempo y eficiencia de codificación, por lo que parecen las opciones más adecuadas para llevar a cabo la regionalización. En el próximo capítulo se expondrán los resultados completos y las conclusiones definitivas a este respecto.

CAPÍTULO 6

RESULTADOS

6. RESULTADOS

6.1 Configuración

El diseño propuesto se ha implementado sobre la versión del software HM13.0 correspondiente al estándar HEVC [Online]. La configuración de los parámetros principales empleada a la hora de realizar las pruebas experimentales, está basada en las recomendaciones dadas en [Bossen, 2011]. De las cuales, las más relevantes son las siguientes:

- Archivo de configuración empleado, *encoder_lowdelay_P_main*.
- 100 planos por cada secuencia de vídeo.
- QP de diferentes valores, 22, 27, 32 y 37.
- Conjunto de secuencias de diferentes resoluciones y un contenido variado de vídeo.
- Patrón IP.

6.2 Reducción de complejidad

Como ya se ha mencionado anteriormente, el objetivo del algoritmo propuesto es reducir la complejidad del proceso de codificación a partir de una decisión prematura respecto a los niveles de profundidad correspondientes a las CUs. De esta manera, se consigue reducir el tiempo de computación empleado en la codificación de una secuencia de vídeo a costa de cierta pérdida de eficiencia en la codificación que se medirá como un aumento del bitrate.

Con el fin de medir de manera precisa las prestaciones del método propuesto, se analizan los siguientes parámetros: incremento del bitrate (ΔBR%), el cual ha sido calculado siguiendo las recomendaciones dadas en [G.Bjontegaard, 2001], y el tiempo de cómputo ahorrado (TS%), el cual se obtiene mediante la siguiente expresión:

$$TS = \frac{Tiempo(HM13.0) - Tiempo(\ método\ propuesto)}{Tiempo(HM13.0)} \times 100, \qquad (6.1)$$

El algoritmo propuesto, ha sido testado en diferentes tipos de secuencias de vídeo, tal y como se puede apreciar en la tabla 6.1. En esta tabla, se incluyen los resultados para los diferentes tipos de regionalización mencionados en el capítulo 5.

Secuencia	Sin Regiones		Cruz		3 regiones horizontales		4 regiones horizontales	
	ΔBR	TS	ΔBR	TS	ΔBR	TS	ΔBR	TS
BasketballDrill 832x480	5.61	40.68	6.04	40.44	6.76	42.11	6.06	41.25
BasketballDrillText 832x480	7.78	41.31	8.21	41.46	7.33	41.34	7.08	40.80
BasketballDrive 1920x1080	6.46	54.63	6.04	52.14	5.56	51.08	4.71	47.81
BasketballPass 416x240	8.44	35.49	8.02	36.41	8.69	38.76	7.54	37.10
BlowingBubbles 416x240	2.94	12.30	1.11	7.76	2.10	12.29	1.80	11.70
BQMall 832x480	3.52	27.68	4.36	27.44	3.44	25.85	4.08	25.64
BQSquare 416x240	1.61	4.66	0.60	3.90	0.61	4.53	0.71	5.45
BQTerrace 1920x1080	1.96	31.25	1.94	27.33	1.51	28.26	1.39	28.09
Cactus 1920x1080	2.79	35.29	3.00	36.08	2.74	36.50	3.19	39.06
Johnny 1280x720	1.64	43.62	1.44	43.57	1.55	47.39	1.23	45.40
Kimono 1920x1080	0.80	19.89	0.58	14.36	0.82	13.53	0.84	14.98
KristenAndSara 1280x720	1.49	43.10	1.92	45.33	1.70	49.67	1.79	49.12
ParkScene 1920x1080	3.91	34.80	0.93	20.79	1.79	23.68	2.58	25.28
RaceHorses 416x240	3.84	9.21	2.10	10.17	2.47	11.37	2.44	11.75
SlideShow 1280x720	9.71	58.07	7.72	58.61	6.41	58.55	5.35	58.24
Vidyo3 1280x720	4.26	58.30	6.38	56.83	4.13	56.10	4.13	53.97
VALOR MEDIO	**4.17**	**34.39**	**3.77**	**32.66**	**3.60**	**33.81**	**3.43**	**33.47**

Tabla 6.1 Análisis del rendimiento del método propuesto considerando diferentes tipos de regionalización

En base a los resultados medios obtenidos de cada una de las diferentes regionalizaciones efectuadas, se puede concluir que tanto las divisiones del plano en 3 y 4 regiones horizontales propuestas, logran reducir la complejidad de manera más eficiente que el resto de opciones contempladas, ya que ambos tipos de división ofrecen prestaciones similares. Si se analiza el valor medio del ΔBR y del TS, el primer aspecto relevante es que por medio de estos resultados, se justifica el uso de diferentes regiones pertenecientes a un mismo plano. Independientemente del tipo de regionalización efectuada, el valor del TS obtenido al dividir el cuadro es aproximado al valor del TS asociado a no establecer ningún tipo de región, y el ΔBR para cualquier tipo de división es inferior a trabajar sin regiones, por lo tanto con las regiones se compromete en menor medida la eficiencia de compresión.

Es importante reseñar que independientemente de la división del plano en 3 ó 4 regiones, a excepción de cuatro secuencias, el porcentaje del tiempo de cómputo ahorrado supera siempre el 20%, mientras que, salvo en cinco secuencias, el porcentaje del incremento del bitrate no supera nunca el 5%. Concretamente las secuencias de *Race Horses* y *Cactus* ofrecen una relación muy eficiente entre el TS y ΔBR. No obstante, secuencias como *BQMall* no aporta los resultados deseados, debido principalmente a que se trata de una secuencia con un alto nivel de complejidad, lo que implica que sea necesario analizar las diversas profundidades de las CU, lo cual evita un incremento significativo del TS.

6.3 Comparación con otros métodos del estado del arte

En este apartado, se realiza una comparación con dos métodos del estado del arte, [Correa et al., 2013] y [Zhang et al., 2013] resumidos en el capítulo 4, en la cual, la configuración de los parámetros de codificación a la hora de realizar las pruebas experimentales, es idéntica respecto a las pruebas anteriores.

En la tabla 6.2 se muestran los resultados obtenidos del ΔBR y del TS para el diseño propuesto y los métodos a comparar. Aclarando que en dicha tabla se incluyen exclusivamente los resultados correspondientes a la división del plano en 3 regiones horizontales, ya que son similares a la división en 4 regiones y, en base a los resultados previos, se han considerado como las mejores opciones.

Secuencia	3 regiones horizontales		Correa DCC 2013		Zhang DCC 2013	
	ΔBR	**TS**	**ΔBR**	**TS**	**ΔBR**	**TS**
BasketballDrill 832x480	6.76	42.11	6.99	25.60	0.97	17.96
BasketballDrillText 832x480	7.33	41.34	6.54	25.51	1.21	18.11
BasketballDrive 1920x1080	5.56	51.08	5.00	29.98	0.97	26.79
BasketballPass 416x240	8.69	38.76	7.67	21.39	0.48	10.06
BlowingBubbles 416x240	2.10	12.29	2.69	16.88	0.03	8.49
BQMall 832x480	3.44	25.85	7.78	23.26	0.56	17.12
BQSquare 416x240	0.61	4.53	3.09	13.67	0.12	8.78
BQTerrace 1920x1080	1.51	28.26	2.84	25.58	0.95	28.76
Cactus 1920x1080	2.74	36.50	3.20	25.95	0.93	29.29
Johnny 1280x720	1.55	47.39	4.01	21.55	4.20	46.96
Kimono 1920x1080	0.82	13.53	0.29	22.75	0.42	24.47
KristenAndSara 1280x720	1.70	49.67	2.09	34.63	14.28	54.15
ParkScene 1920x1080	1.79	23.68	3.38	27.24	1.31	24.89
RaceHorses 416x240	2.47	11.37	5.30	15.72	0.07	7.78
SlideShow 1280x720	6.41	58.55	6.68	28.53	42.66	51.30
Vidyo3 1280x720	4.13	56.10	2.04	30.92	3.05	42.94
VALOR MEDIO	**3.60**	**33.81**	**4.35**	**24.32**	**4.51**	**26.11**

Tabla 6.2 Análisis del rendimiento del método propuesto en comparación con [Correa et al., 2013] y [Zhang et al., 2013]

Observando los resultados medios de los tres métodos, no cabe duda de que el método propuesto reduce la complejidad de una manera mucho más eficiente, puesto que en lo referente al TS, logra casi un 10% más respecto a [Correa et al., 2013] y supera en un valor cercano al 8% a [Zhang et al., 2013], teniendo en cuenta que el ΔBR producido es menor, en comparación con los resultados generados por los otros dos métodos.

Siendo más concretos, en la comparación con [Correa et al., 2013], exceptuando cinco secuencias, para el resto de ellas el método propuesto supera con creces el TS originado por [Correa et al., 2013], resaltando que en la mayoría de estos casos, el ΔBR es claramente inferior respecto al método a comparar.

Al realizar un estudio más profundo de los datos obtenidos, destacan los resultados correspondientes a la secuencia *Johnny*. En esta secuencia el algoritmo desarrollado supera el doble de TS, empleando para ello menos de la mitad del ΔBR en comparación con [Correa et al., 2013].

En relación con [Zhang et al., 2013], en una gran parte de los casos presentados, el método propuesto alcanza resultados considerablemente superiores en cuanto a TS, sin embargo para lograr estos resultados genera mayores incrementos de bitrate. Este resultado es lógico puesto que tal diferencia de TS es razonable que genere mayores pérdidas de eficiencia.

Si se analiza los resultados con más detalle, llaman la atención los datos obtenidos para el vídeo *Slides Show*. En lo que concierne a esta secuencia, el método propuesto es capaz de generar un mayor TS si lo comparamos con [Zhang et al., 2013], no obstante el dato más relevante reside en que consigue estos resultados produciendo un 36% menos de ΔBR que el método a comparar. Esta secuencia consiste en un gran número de cambios de escena, por lo que cualquier método no adaptativo, sufre grandes pérdidas, siendo éste un inconveniente superado por el diseño propuesto.

Sin embargo, para la secuencia *Kimono* los resultados no son satisfactorios, ya que el método propuesto genera mayor ΔBR y ofrece peores resultados en relación al TS en comparación con los métodos presentados en [Correa et al., 2013] y en [Zhang et al., 2013].

Para ilustrar la eficiencia de codificación de los métodos comparados en este apartado, se muestra el rendimiento en forma de curvas RD en las figuras 6.1 y 6.2 para la secuencia *Johnny,* mientras que las figuras 6.3 y 6.4 corresponden a la secuencia *Park Scene*.

Las figuras 6.1 y 6.3 representan las curvas RD completas, mientras que en las figuras 6.2 y 6.4 se muestran zonas específicas de estas curvas, para poder observarlas con mayor detalle. Como se puede observar, el método propuesto se aproxima más al software de referencia que los métodos [Correa et al., 2013] y [Zhang et al., 2013], es decir, que para un mismo valor de bitrate, el diseño propuesto produce menores pérdidas en la calidad visual, indicada en dichas gráficas como PSNR (*Peak Signal to Noise Ratio*).

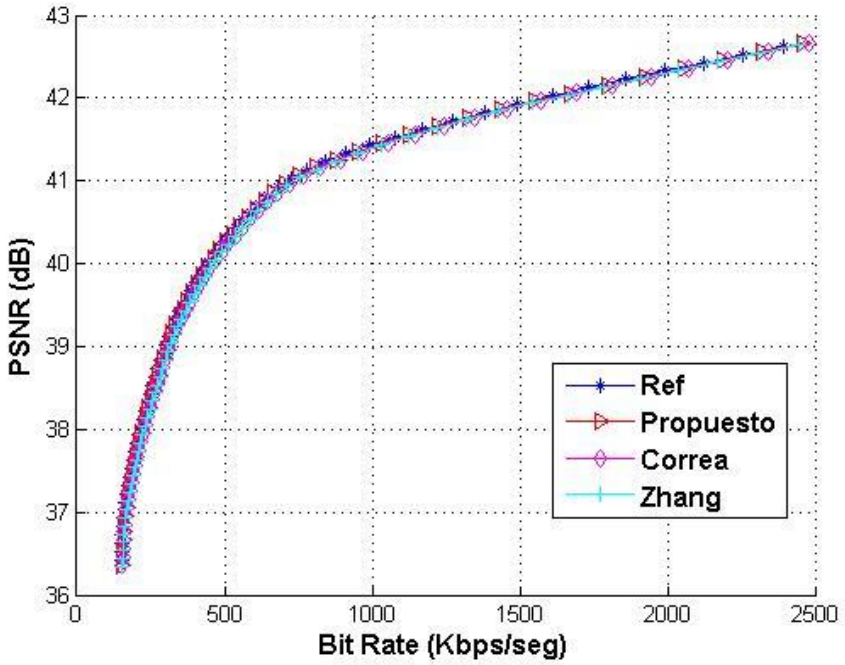

Figura 6.1 Rendimiento RD para la secuencia *Johnny*

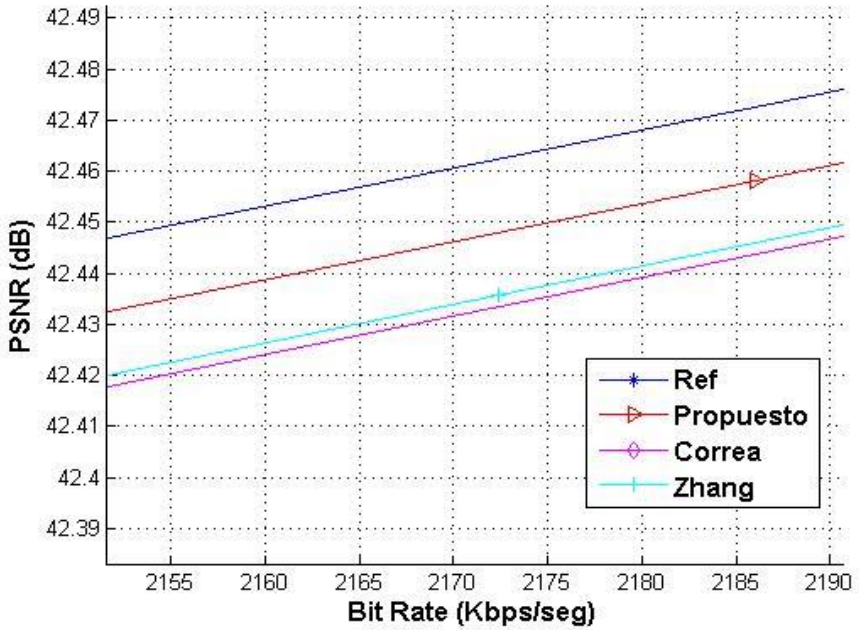

Figura 6.2 Zona ampliada de la figura anterior

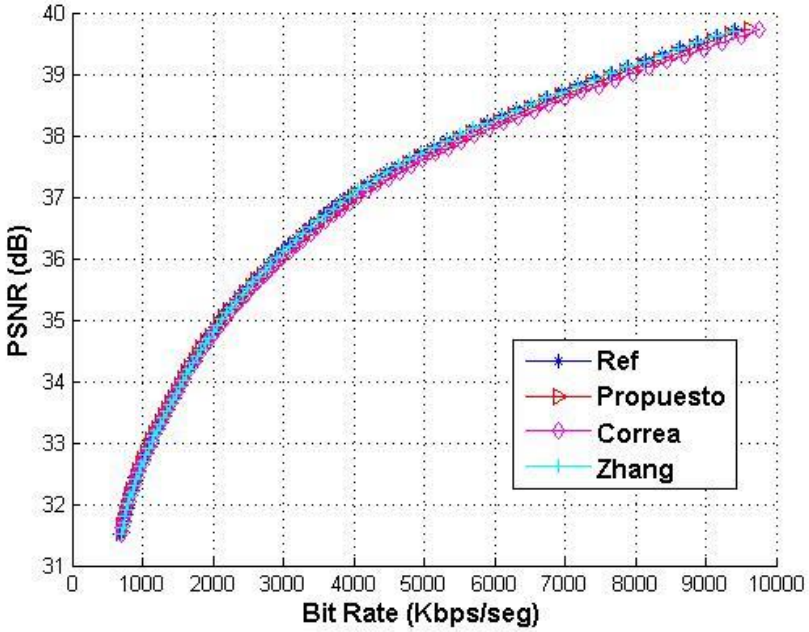

Figura 6.3 Rendimiento RD para la secuencia *Park Scene*

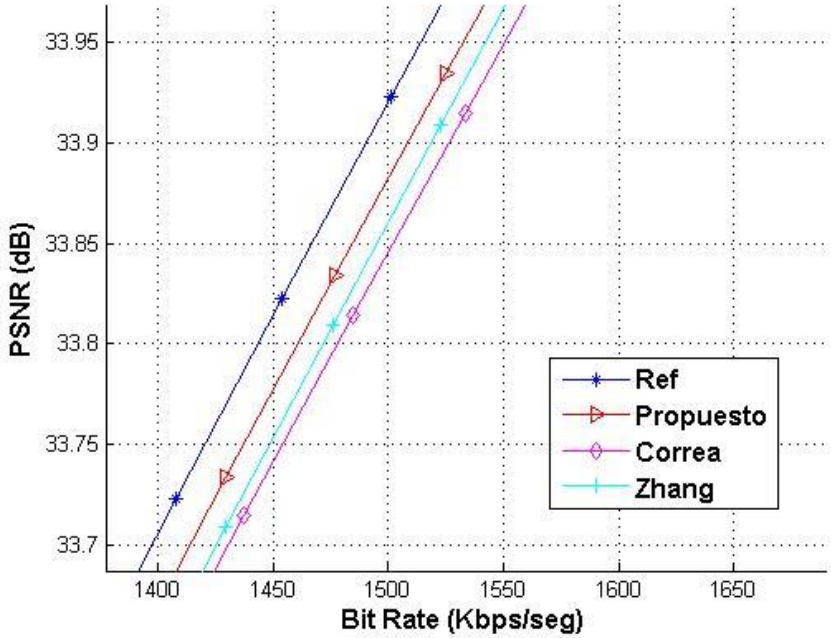

Figura 6.4 Zona ampliada de la figura anterior

CHAPTER 7

CONCLUSIONS AND FUTURE LINES OF RESEARCH

7. CONCLUSIONS AND FUTURE LINES OF RESEARCH

7.1 Conclusions

This project has shown an algorithm, which main goal is to reduce complexity during the encoding process made for the newest video standard, HEVC. On the one hand, this standard reaches a high capacity of compression when applying new technics, for example, building the prediction using a tree structure. On the other hand, these techniques increase the complexity of the diverse involved encoding processes.

The proposed method aims to reduce the computing time used in encoding a video sequence without changing significantly its visual quality. Furthermore, this method defines an earlier decision of the different depths levels that are related with coding basic units that form a frame. This is because this process requires one of the biggest computing loads. One of the main improvements of our proposal respecting other systems reviewed in the state of the art consists of considering diverse regions in the frames, since it is often found very different characteristics between different regions of the scene. Thus a statistical analysis is done independently for each of them.

Another essential point of this algorithm is that the evaluation of the entire statistics features that define the decisions boundaries are carried out online. This means that the performance of the proposed method increases thanks to its capability of adapting to the variable's content in the video sequences time.

Diverse tests have been taken comparing the obtained results using different types of regionalization, as well as with other methods of the estate of the art, proving that it can reach higher results than other proposals to proof the efficiency of this system.

The main features of designed method are the following:

1. Dividing the frame in different regions in order to stimulate the correlation of data in the same region avoiding data accumulation between regions with very different features.
2. Using a hypothesis test to decide whether a CU should be divided or not.
3. Determinate RD costs as input features of the hypothesis test.
4. Assuming the Gaussian distribution in order to simplify the process.
5. An online estimation of the statistics features in order to adjust the thresholds in a dynamic way and thus, adapting video contents. The main goal of this action is to achieve a higher efficiency when decreasing complexity.

This algorithm accomplishes the following requirements:

1. It only entails a low computing cost.
2. It is compatible with diverse video resolutions and very varied content.
3. It provides excellent results to a great variety of sequences, reaching reasonable time computing saves and generating with it few coding efficiency loses.
4. Most of the time, the results improves the performances of other methods of state of the art.
5. Capability of adapting to different features of video sequences.

7.2 Future lines of research

There are different possible types of future research guidelines for this work.

The first one is related to modify the unidimensional hypothesis test towards a multidimensional one instead. With this change, it would be possible to consider more input features. Consequently, it would lead to carry out a deeper study of video characteristics improving the performance of the hypothesis test used for this case.

Secondly, it is possible to consider to extend the model of this algorithm to a deeper levels related to PUs and TUs. This is due to the different units used by HEVS during the coding process. The main goal is to reduce complexity as much as possible.

Other possible future guideline is to use stronger decision – making systems, like for example neuronal networks and support vector machines, since these algorithms would allow obtaining a higher precision of premature decision.

Finally, instead of dividing the frame between different regions, it is feasible to separate concrete objects that currently exists in each plane. Thereby, it would be possible to increase the spatial correlation, allowing executing better decisions.

CAPÍTULO 8

PLANIFICACIÓN Y PRESUPUESTO

8. PLANIFICACIÓN Y PRESPUESTO

8.1 Planificación

En este apartado se desarrollan las diferentes etapas que se han llevado a cabo para la realización del trabajo.

1. Documentación. Hace referencia a la recopilación de información acerca de los diferentes procesos que se llevan a cabo en la codificación de vídeo, del estándar HEVC y de la búsqueda de artículos relacionados con el estado del arte entre otros. El tiempo empleado para esta labor ha sido de 100 horas.
2. Análisis del código. Esta etapa está relacionada con el estudio del código correspondiente al estándar HEVC. Su duración ha sido de 60 horas.
3. Implementación. Dicha etapa corresponde al desarrollo del software asociado con el algoritmo propuesto. Al ser la etapa de mayor importancia, se han empleado 200 horas en ella.
4. Pruebas experimentales. En esta etapa se han llevado a cabo una variedad de pruebas con el fin de comprobar el correcto funcionamiento del método diseñado. El tiempo destinado a la etapa en cuestión, equivalen a 100 horas.

Al tener en cuenta todas las etapas involucradas en el desarrollo global del trabajo, el tiempo total dedicado a su elaboración asciende a 410 horas.

8.2 Presupuesto

A continuación se detalla un estudio económico asociado a la elaboración de este trabajo. El estudio se divide en dos partes: coste del personal y coste de los componentes hardware y software.

En la siguiente tabla de desglosan los costes relacionados con el personal:

Nombre	Puesto	Euros/Hora	Horas totales	Coste
Roberto Aldeán	Ingeniero técnico	20€	410	8200€
Amaya Jiménez	Ingeniero superior	35€	100	3500€
Subtotal				**11.700€**

Tabla 8.1 Coste del personal

En la siguiente tabla se desglosan los costes de los diferentes equipos y software empleados:

Producto	Coste
Ordenador portátil TOSHIBA Satellite L750/L755	500€
Software	400€
Subtotal	**900€**

Tabla 8.2 Coste de los componentes hardware y software

Por lo tanto, considerando el coste relacionado con el personal y el coste correspondiente a los componentes hardware y software utilizados, el coste total del proyecto es de **12.600€**

CAPÍTULO 9

REFERENCIAS

9. REFERENCIAS

[Bossen, 2011] F, Bossen. *JCTVC-F900 Common test conditions and software reference configurations*. Fifth meeting of the Joint Collaborative Team on Video Coding (JCT-VC) Torino, IT, 2011.

[Cho et al., 2013] Seunghyun Cho and Munchurl Kim. IEEE transactions on circuits and systems for video technology, vol.23, no. 9, pp. 1555-1563 , September 2013

[Correa et al., 2013] G. Correa, P. Assuncao, L. Agostini, and L. Cruz. *Coding Tree Depth Estimation for Complexity Reduction of HEVC*. Data Compression Conference (DCC), pp. 43–52, March 2013.

[Cover and Thomas, 2001] Thomas M. Cover and Joy A. Thomas, 2001. *Elements of Information Theory*.

Damián Ruiz Coll. Presentación: El nuevo estándar de codificación de vídeo H.265

[Everett, 1963] H, Everett. *Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources*. Operations Research, 11(3):399-417, 1963.

[G.Bjontegaard, 2001] G.Bjontegaard. *Calculation of Average PSNR Differences Between RD-Curves*. ITU-T, VCEG-M33, Apr. 2001

[Grellert et al., 2013] Mateus Grellert, Muhammad Shafique, Muhammad Usman Karim Khan, Luciano Agostini, Julio C. B. Mattos and Jörg Henkel. *An adaptative workload management scheme for HEVC encoding.*

[Hsu et al., 2013] Wei-Jhe Hsu and Hsueh-Ming Hang. *Fast Coding Unit Decision Algorithm for HEVC*. Department of Electronics Engineering, National Chiao-Tung University, Hsinchu, Taiwan

I. Richardson , 2003. *H.264 and MPEG-4 video compression.*

Iván González Díaz. Transparencias de la asignatura: Ingeniería de sistemas de vídeo.

[Jiménez-Moreno et al., 2016] Amaya Jiménez-Moreno, Eduardo Martínez-Enríquez, and Fernando Díaz-de-María. *Complexity Control Based on a Fast Codgin Unit Decision Method in the HEVC Video Coding Standard*. IEEE transactions on multimedia.

[Kim et al., 2013] Jongho Kim, Yoonsik Choe, and Yong-Goo Kim. *Fast Coding Unit Size Decision Algorithm for Intra Coding in HEVC*. 2013 IEEE International Conference on Consumer Electronics (ICCE).

Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin. Han, and Thomas Wiegand. *Overview of the High Efficiency Video Coding (HEVC) Standard*. IEEE Transactions on circuits and Systems for video technology, vol 22, no 12, pp. 1649-1668, December 2012.

Miguel Ángel Bona San Vicente. Transparencias de la asignatura: Televisión digital.

[Online] High Efficiency Video Coding. Available: http://hevc.hhi.fraunhofer.de/.

[Shen et al., 2012] Xiaolin Shen, Lu Yu and Jie Chen. *Fast Coding Unit Size Selection for HEVC based on Bayesian Decision Rule*. 2012 Picture Coding Symposium.

[Shen et al., 2013] Liquan Shen, Zhi Liu, Xinpeng Zhang, Wenqiang Zhao, and Zhaoyang Zhang. *An Effective CU Size Decision Method for HEVC Encoders*. IEEE Transactions on multimedia, vol 15, no 2, pp. 465-470, February 2013.

[Sullivan and Wiegand, 1998] G, Sullivan and T.Wiegand. *Rate-distortion optimization for video compression*. Signal Processing Magazine, IEEE, 15(6):74-90, 1998.

[Zhang et al., 2013] Y. Zhang, H.Wang, and Z. Li. *Fast coding unit depth decision algorithm for interframe coding in hevc*. Data Compression Conference (DCC), pp. 53–62, March 2013.

[Zhao et al., 2011] Liang Zhao, Li Zhang, Siwei Ma and Debin Zhao. *Fast mode decision algorithm for intra prediction in HEVC*. November 2011.